Call-A.P.P.L.E.™

Volume VI, Number 5 May 1983 $2.50 ($3.00 Canada, £1.75 UK.)

SOFTWARE MARKETING:

You Think You Have a Million Dollar Program

Page 9

HANDLING LARGE PASCAL PROGRAMS:

Getting a Quart into a Pint Pot

Page 13

A Call -A.P.P.L.E. APPLE-CATION:

Magfiler

Page 19

YOUNGBLOOD:

A Complete Password System

Page 27

Apple PugetSound Program Library Exchange

TRUNKS FOR THE MEMORIES.

Introducing the most logical place to store Elephant Memory Systems® (or lesser brands of disks): The Trunk.

With its alphabetized library index, you can file or retrieve up to 60 disks, instantly.

The Trunk is made of durable molded plastic with a hinged, one-piece lid, to keep disks safe from dust, dirt, and other detriments which disks despise.

And, it's portable. Because the lid doubles as a carrying handle so your Elephant Memory Systems® disks can go anywhere you do.

There's a model for 5¼″ and 8″ floppies, as well as a cassette-and-game file and a special Atari® version.

So if you're looking for the best disk storage system on the market . . .

The Trunk is an open-and-shut case.

THE TRUNK. ENDORSED BY ELEPHANTS.

Elephant Memory Systems® Disks

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a life-time of heavy-duty use).

Marketed exclusively by Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021. Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.

Circle 18 on Reader Service Card

VistaDaVinci
GREAT INNOVATORS

15th century renaissance man, Leonardo Da Vinci, contributed an inexhaustible collection of inventions and ideas to solve the world's problems. These ideas were centuries ahead of their time. The studies on this page deal with experiments in manned flight.

Like Leonardo, Vista Computer is answering many of today's complex computer storage problems. The Vista V1200 is a great solution to Apple II™ storage. Mass storage for your Apple II™ Computer has always been a problem. On one hand, there were the exotic, expensive hard disks with no cost efficient means of backup. On the other hand, the Apple floppy drive lacked the speed and storage demanded by today's professionals.

Vista's V1200 offers both at an incredibly attractive price. The removable VistaPak cartridges offer 6 Megebytes of removable storage each and can be backed up like a floppy.

Now hard disk storage and speed can be yours with the added capability of interchangeable media. The V1200 eliminates the worries of head crashes, drive alignments, lost data, or backup with a new application of field-proven floppy technology.

The VistaPak cartridges hold 6MB of formatted data each. The removable cartridge allows you to keep duplicates of your valuable data as well as to keep separate paks for your accounting, word processing, spread sheet and other applications. No other storage device offers more in flexibility and capability.

• Microprocessor controlled drive • DMA Controller • Removable Data Cartridges • CP/M, DOS & Pascal compatible • Quickcharge™, DOS enhancement included • Includes 1 VistaPak cartridge • Vista 120 Day Warranty

Vista V1200

™Apple II is a registered trademark of Apple Computer Co.

Contact Your Local Vista Dealer or Call our Vista Hotlines.

Vista COMPUTER COMPANY, INC.

1317 East Edinger / Santa Ana, CA 92705
(714) 953-0523 / (800) 854-8017

DISTRIBUTORS / REPRESENTATIVES

Western - Group 3 Wholesale
(213) 973-7844 (408) 732-1307

South Central - M.P. Systems
(214) 385-8885

Northeast - Computers & Peripherals Inc.
(315) 476-6664

Circle 40 on Reader Service Card

APPLE PUGETSOUND PROGRAM LIBRARY EXCHANGE:

CLUB INFORMATION

GENERAL OFFICES
21246 68th Ave. S.
Kent, WA 98032 — (206) 872-2245
Hours: 9 - 4 Monday - Friday
Members Services/Dealer Orders

EDITORIAL DEPARTMENT
5606 California Ave. S.W.
Seattle, WA 98136—(206)932-8288
Hours: 8-4 Monday - Friday

ADVERTISING REPRESENTATIVES
Grawin Publications
1020 Lloyd Building
Seattle, WA 98101 (206) 223-0861

MEETING NOTICES

7:30 PM, Tuesday, May 17, 1983
A.P.P.L.E. Office
21246 68th Ave. S., Kent
Speaker: Wm. R. Miller
"3-D Graphics"

7:30 PM, Tuesday, June 21, 1983
A.P.P.L.E. Office
21246 68th Ave. S., Kent
Speaker to be announced

7:30 PM, Tuesday, July 19, 1983
A.P.P.L.E. Office
21246 68th Ave. S., Kent
Speaker to be announced

Call —A.P.P.L.E. HOTLINE

Technical/programming assistance only

DAYTIME HOTLINE: (206) 872-9004
9 AM - 3 PM Only
Mon. - Fri. Only

EVENING HOTLINE: (206) 932-6588
6 PM - 10 PM
7 days per week

WEEKEND HOTLINE: (206) 932-6588
9 AM - 3 PM
6 PM - 10 PM
Sat. - Sun.

All times shown are PACIFIC time.

Membership Information

Please direct telephone questions on membership and orders to (206) 872-2245.

One Time Apple—Cation Fee	$25.00
Annual Dues	$20.00
Total	$45.00

Additional postage required for foreign memberships and shipments.

Additional membership postage:

- Canada, West Indies, Central America, Columbia, Venezuela and Mexico — Add $15.00 ($60.00 total)
- All S. American countries (except above), Europe and N. Africa — Add $21.00 ($66.00 total)
- Asia, Mid East, S. Africa, Australia and Pacific Islands — Add $27.00 ($72.00 total)

All orders must be prepaid in U.S. funds only.

Membership is on an annual, 12 month basis, commencing with the month of enrollment. Upon receipt of total membership fees, new members will receive an Apple—Cation blank, order form and description sheet of club products.

Please allow up to 30 days for a turnaround on membership and orders.

Call —A.P.P.L.E. is mailed free twelve time per year as benefit of membership. Subscriptions are not available.

1983 back issues available at $2.50 each.

Call —A.P.P.L.E. bound volumes for previous years are available as indicated on the member order form on the last page.

Institutional memberships are now available. Contact General Office for details.

Prospective authors please send a large self-addressed stamped envelope (37¢ postage) to Editorial Dept. for a copy of Author's guidelines.

Cover and interior design by Vic Warren Design.

A.P.P.L.E. on the SOURCE

A special section of The Source is open only to A.P.P.L.E. members who may download Call -A.P.P.L.E. programs. To access, type:

CALL-APPLE from Source command mode. To register, send mail to ST1570 with your name, membership number, expiration date and Source account number.

Changes of address must be received at the General Office at least 30 days prior to effective date of move. Issues missed due to non-receipt of change of address notification must be separately reordered by the member at current cost. Change of address form may be found on page 39 of this issue.

A.P.P.L.E. CRATE BULLETIN BOARD

(206) 935-9119
(usually 24 hours)

Officers and Directors

Dick Hubert, President	(206) 271-6939
Bob Huelsdonk, Vice President	(206) 362-4910
Dave Lingwood, Secretary	(206) 244-9360
Bob Clardy, Treasurer	(206) 641-1917
Val J. Golding, Director	(206) 932-6588
Ken Smith, Director	(206) 322-4199
Don Williams, Director	(206) 242-6807

Office Staff

Dick Hubert	Executive Director
Don Steele	Assistant to Executive Director
Dave O'Leary	Controller
Sally Davies	Manager of Operations
Arliss Sword	Dealer Orders
Patti Hoffmann	Special Projects Coordinator
Gunter Hirt	Bookkeeping, Member Records
Joff Morgan	Technical Operations
Cheryle McCalmant	Order Processing
Kris Mortenson	Order Processing
Sharon Birdsell	Order Processing
Gayle Rank	Secretary/Receptionist
Mike Christensen	Systems Development
James Redhed	Receiving
Brad Phillips	Shipping/Receiving
Steve Johnson	Technical Staff Manager
Steve Ross	Software Product Development
Garry Reinhardt	Systems Development
Butch Greathouse	Technical Support
Kevin O.Donohoe	Manager of Marketing

Magazine Staff

Val J. Golding	Editor	(206) 932-8288
Kathryn Hallgrimson	Associate Editor	(206) 932-8288
Kathy Pride	Editorial Assistant	(206) 932-8288
Jonathan V. Post	Contributing Editor	(206) 824-7641
Bruce van Norman	Pascal Editor	(206) 932-8288
Jim Hockenhull	Staff Writer	(509) 334-4137
Clifton M. Howard, M.D.	Staff Writer	(201) 768-9274
Paul Mosher, M.D.	Staff Writer	(518) 765-4118
Bill Parker	Staff Writer	(619) 297-9343
Lee Reynolds	Staff Writer	(305) 722-2639
Clay Ruth	Staff Writer	(219) 865-6165
David G. Sparks	Staff Writer	(319) 232-2313
Ralph H. Swerdlow, M.D.	Staff Writer	(916) 920-8709
Grawin Publications	Production, Advertising	(206) 223-0861

A.P.P.L.E. consultants will be found on page 79 of this issue.

Call-A.P.P.L.E.™

Volume VI, Number 5 May 1983/Menu

Val J. Golding, Editor
Kathryn Hallgrimson Suther, Associate Editor

ENTRÉES

9 SOFTWARE MARKETING:
You Think You Have a Million Dollar Program
Alan D./Valerie A. Floeter

13 HANDLING LARGE PASCAL PROGRAMS:
Getting a Quart into a Pint Pot
Chris W. Green

19 A Call -A.P.P.L.E. APPLE-CATION:
Magfiler
Jack Olivieri

27 YOUNGBLOOD:
A Complete Password System
Jason K. Mintz

À LA CARTE

35 A Call -A.P.P.L.E. TECHNOTE:
Applesoft Can Throw Your Valuables in the Trash Can
Cornelis Bongers/Call -A.P.P.L.E. staff writer

37 AN ECONOMIC AID:
Automated Typesetting
Mike Shadick

39 A Call -A.P.P.L.E. PRODUCT REVIEW:
Window
Michael Thyng/reviewer

43 A Call -A.P.P.L.E. SOFTWARE REVIEW:
Special Effects
Jeff Harper/reviewer

45 APPLEMASH:
Bored with Bulletins?
Michael Thyng/Call -A.P.P.L.E. staff writer

47 **Double Monitors for the Apple][**
Greg DeWilde

51 **Spelling Corrector**
Wesley R. Felty

57 A Call -A.P.P.L.E. PRODUCT REVIEW:
VB-3 Microkeyer
James L. Hockenhull/reviewer

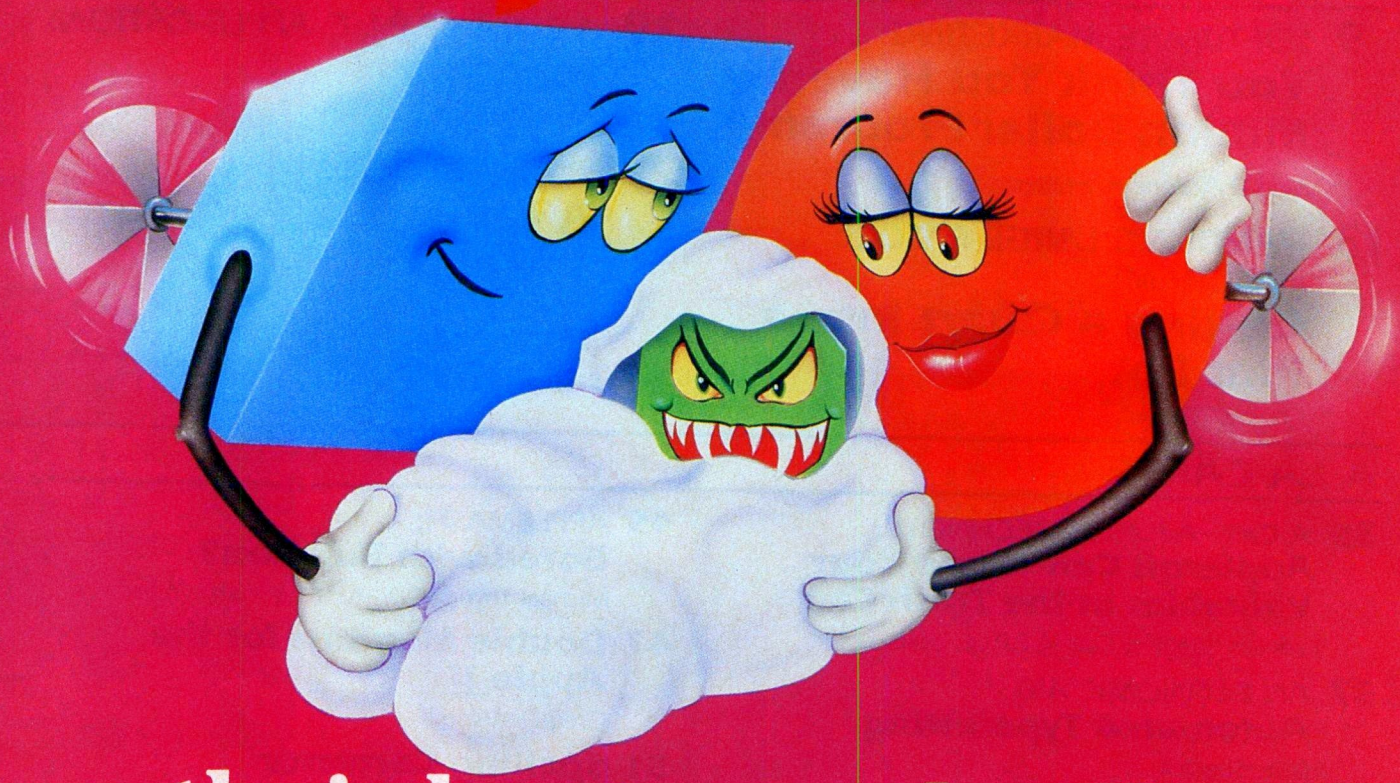
HORS d'OEUVRES

2 A.P.P.L.E. Club Information
5 The Editor Bytes Back
the editor
7 Bytes from A.P.P.L.E.
Val J. Golding
7 BOOT
Craig MacIntosh
61 Write -A.P.P.L.E.
the readers
67 Tomorrow's Apples Today
Kevin O. Donohoe/editor
71 Writing the Wrongs
the wrongdoers
74 A.P.P.L.E. INFO:
Pacific Science Center Festival
Michael Thyng/Call -A.P.P.L.E. staff writer
75 A.P.P.L.E. Doctor
David G. Sparks/Call -A.P.P.L.E. staff writer
79 Call -A.P.P.L.E. Consultants
79 A.P.P.L.E. Order Form
80 Call -A.P.P.L.E. Unclassifieds

Apple user groups may reprint without prior permission any portion of the contents herein, provided author, title, and publication credits are given and a written notice of such use furnished to the editorial office. For commerical applications of programs herein, the author should be contacted c/o General Office. Call —A.P.P.L.E. is published monthly by Apple Puget Sound Program Library Exchange, 21246 - 68th Ave. S., Kent, WA 98032. Subscriptions are not available but copies are mailed free as a benefit of A.P.P.L.E. membership.

Entire contents copyright© 1983 by
Apple PugetSound Program Library Exchange

When these aliens get personal—



their brats get deadly

You've never had more charming, lovable aliens residing on your screen. Different shapes, different colors—and thoroughly friendly . . . until they meet in the MATING ZONE. Once they do, they produce offspring who are genetically mean and ornery. And their mission in life is to take yours!

At each level you find different aliens who have this great compulsion to produce mutated offspring—and it becomes easier for them, but more difficult for you, as the number of Mating Zones increase at each level!

MATING ZONE™

For the most intriguing, enjoyable game of the year—visit the MATING ZONE and try to stop the population explosion!

$29.95 for the Apple II+*

At your computer store, or:

DATAMOST™

8943 Fullbright Ave.,
Chatsworth, CA 91311
(213) 709-1202

VISA/MASTERCARD accepted.
$2.00 shipping/handling charge. (California residents add 6½% sales tax.)

*Apple II+ is a trademark of Apple Computer, Inc.

Copyright 1983 Datamost Inc.

Circle 49 on Reader Service Card

THE EDITOR BYTES BACK

the editor

MAY introduces the team of **Alan and Valerie** (nice name!) **Floeter** to Call -A.P.P.L.E. readers. We often have received hotline calls asking how to market a program the caller had just written; now comes **Million Dollar Program** to answer those queries.

If you are a dyed-in-the-wool Call -A.P.P.L.E. reader, then **Jack Olivieri's Magfiler** Apple-cation will help you out; a complete system to index all your Call -A.P.P.L.E. articles, and those from (ahem) other magazines, as well.

Pascal readers will dig **Chris Green's Getting a Quart into a Pint Pot,** a treatise on getting more diskette room when there ain't none!

Fifteen year old **Jason Mintz** is also a newcomer to our pages, but don't let his age throw you. He read **Ralph Swerdlow's Password Protection** in a previous issue and wrote **A Complete Password System** as an outgrowth. It is both complete and comprehensive.

In **A Call -A.P.P.L.E. TECHNOTE,** staff writer **Cornelis Bongers** has uncovered a major Applesoft garbage collection bug, which is detailed in **Applesoft can Throw your Valuables in the Trash Can.** As it turns out, if garbage is forced during the string assignment process, strings can be entirely lost. Try his demo program and see for yourself.

Staff writer **Jim Hockenhull,** whose specialty is video interfacing, reviews the **VB3 Microkeyer** in this issue, while **Jeff Harper** takes a look at **Special Effects,** a neat high-res utility.

Mike Shadick, one of our more prolific authors, returns this issue with some info on how **Automated Typesetting** can be an economic aid in these pressing times. An accompanying feature describes how **Call -A.P.P.L.E.** helped pioneer the field.

We recently have received a number of questions on random number generation, and since **Dave Sparks** recently pointed out the flaws in Applesoft's random number generator, we forwarded the questions to him. The results can be seen in this month's **A.P.P.L.E. Doctor.**

We hope to introduce two new columns next month: **CP/MAGIC** by consultant **Murray Arnow** will focus on some Apple CP/M tricks and techniques, while **The Northern Spy,** authored by newly appointed staff writer **Rick Sutcliffe** will present some north-of-the-border views and patter. Rick will also be supporting his new A.P.P.L.E. product **Amper-Manager.**

Educators should also take note: Next month will feature an **educational software** section, including reviews and general material on the field. Particular emphasis will be placed on the concept of educational arcade games. Tune in and find out what it's all about.

Circle 27 on Reader Service Card

☐ VISISCHEDULE ☐ SUPERCALC ☐ VISICALC ☐ WORDSTAR ☐ D.B. MASTER ☐ MULTI PLAN ☐ VISIFILE ☐ dBASE II ☐

RENT SOFTWARE BEFORE YOU BUY!

from our

SOFTWARE RENTAL LIBRARY

You can now RENT the most popular software available for just 20-25%* of Manufacturers' Retail Price

- Eliminate the risk—rent first!
- 100% of rental fee applies toward purchase
- All purchases are 20% Off of Manufacturer's Suggested List
- Rentals are for 7-days (plus 3 days grace for return shipping)
- No Membership Fees

Now currently available for:

Apple	IBM, PC	Standard CP/M
Eagle	TRS-80 II	Xerox 820
Northstar	Osborne	Heath/Zenith 89
	Franklin	

REMEMBER, THESE ARE NOT DEMOS, BUT ORIGINAL UNRESTRICTED SOFTWARE PROGRAMS

(complete with manuals in original manufacturers' packages)

To Immediately Order, or for more information:

UNITED COMPUTER CORP.
Software Rental Library
Culver City, California

Toll Free CALL 1-800 992-7777
In California CALL 1-800 992-8888
In L.A. County CALL 1-213 823-4400

MasterCard AMERICAN EXPRESS VISA

*Plus postage and handling.

☐ AZTEC ☐ SUBMARINE COMMANDER ☐ WIZARDRY—SCENARIO I ☐ PREPPIE ☐ CANYON CLIMBER ☐ DEADLINE ☐ STAR RAIDERS ☐ CENTIPEDE ☐ ZORK II

☐ VOLKSWRITER ☐ DATA PERFECT ☐ FILE MANAGER 80+ ☐ SCREENWRITER PROFESSIONAL ☐ PFS: GRAPH ☐ THE SENSIBLE SPELLER ☐ VISITREND/PLOT

☐ FROGGER ☐ CHOPLIFTER ☐ GORF ☐ DAVID'S MIDNIGHT MAGIC ☐ EASTERN FRONT (1941) ☐ ZORK I ☐

BYTES FROM A.P.P.L.E.

Val J. Golding

HARDWARE again leads our column this month. The *Prowriter 8510A* printer was featured in our March products catalog, which was mailed about the 10th of March to every member of record at that time. Priced at $380, it is a tremendous steal.

We have also added two new disk drives to our line; this acquisition was too recent to make the products catalog, but look at these prices and performance: The *A.P.P.L.E. Silentdrive/slim* and *Silentdrive/dual* at $285 and $500 respectively. Both drives are quieter and faster than any standard drive on the market, carry a full year manufacturer's limited warranty and have been thoroughly checked out in our lab. The slim is less than two inches high, and the dual is two drives in a single cabinet, together the size of one drive.

The specials flyer which accompanied our products catalog also offered some sensational bargains; the prices are good through the end of May, so better hurry. Here is just a sampling:

- *USI green screen monitor* with 20mhz bandwidth at $90 (quantities are limited).
- The $240 *Mockingboard* special, includes the popular Mockingboard and two items of software designed to work with the Mockingboard, *Microbe* by Synergistic and *Thunderbombs* by Penguin. The Mockingboard version of these two programs may also be ordered separately at $45.
- At an $1195 introductory price, the *Vista V-1200* is a state-of-the-art floppy disk cartridge storage device that can handle up to six megabytes of data, and easy to back up.

Flash! As we go to press with this column, we have received a price reduction from our *Mockingboard* supplier which we are passing on to our members: the new price for the above $240 special is now $200. Here's one you can't afford to pass by.

THE Spreadsheet version 2.0 was an instant success at the West Coast Computer Faire with its built in 70 column mode and 80 column card compatibility. It works perfectly with the Apple *IIe* as well as the standard Apple][plus. Since it can build models up to 512k (using four 128k cards) and read DIF files, the member price of $60 can't be beat anywhere. Add the $45 to join A.P.P.L.E. and you are still money ahead, to say nothing of having our library of sophisticated software and hardware discounts available to you.

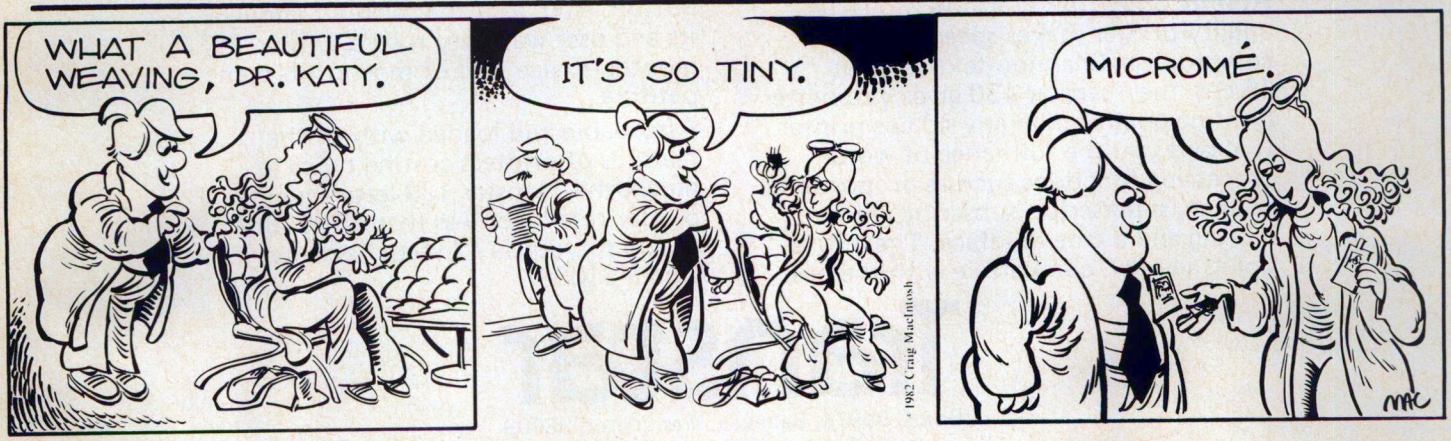
We saved the best for last. At the computer shows we have been giving away with membership our three bound volumes of *Call -A.P.P.L.E.* for 1978, 1979 and 1980, a $36 value. A number of members have written and called to ask why just the new members get the deal — after all, one of the reasons they joined A.P.P.L.E. was for the savings we offer *members,* so here is our peace offering: Send us $20 and we will ship you volumes 1, 2 and 3 of *PEEKing at Call -A.P.P.L.E.* — that's almost 50% off! Ask for the *PEEKing special.* Secondly, If you have very recently joined, we'll offer you our annuals for 1981 and 1982. The regular member price is $18.50 each, and we'll make this pair available at just $25, which represents a savings of about ⅓. We'll call this the *Annual Special.* These two specials are good until June 30, 1983 and represent your most cost-effective way to obtain all the valuable information we have published over the years. Foreign members should note that additional overseas postage is required for both of these specials. Exact amounts are specified in an advertisement elsewhere in this issue.

There is just a whole bunch of exciting new software in the offing, including *Amper-Manager,* which was reviewed in the March issue, but we're running out of space, and will have to leave that for another time.

Hasty la vista!

BOOT

Craig MacIntosh



Is 42 seconds worth $1000?

The new Transtar 130 daisy wheel printer generates a full-page letter in 78 seconds. The least expensive 40 cps printer does it in 36. Only 42 seconds difference...for twice the price.

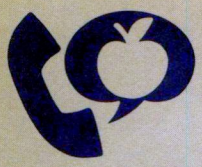
At only $895, the Transtar 130 letter-quality printer makes speed its only compromise. Shannon-text rated at 16 cps, the Transtar 130 gives you better printing quality than any 40 cps printer. It allows you the full range of word-processing functions such as proportional spacing, superscript, subscript, underscoring and a true boldface. The 130 is "plug and go" compatible with the best-selling word processing packages. It's quiet: only 65dB. It's durable. It boasts a unique new autoload feature that automatically loads paper to one of four preselected positions with the touch of a button. And, as if that weren't enough, its end-user warranty runs a full six months—twice that of most of its competitors.

Affordable and loaded with all the features of printers costing twice as much, the Transtar 130 letter-quality printer retails for less than $900. But it's up to you: Is 42 seconds really worth $1000?

Transtar

Box C-96975, Bellevue, Washington 98009

Circle 5 on Reader Service Card



MORE INFORMATION about the products and services in this issue FREE!

Just circle the number and fill out the card. (Note: card expires three months following month of publication.)

READER SERVICE CARD

Free Information on the items circled

Card Expires July 31, 1983 (5-83)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210

1. What kind of product(s) are you likely to buy in the next three months?

A. ☐ Hard Disk
B. ☐ Increased Capability Floppy Disk
C. ☐ Apple Compatible Floppy Disk
D. ☐ Computer
E. ☐ Printer
F. ☐ Modem
G. ☐ CRT

2. More information on Apple PugetSound Program Library Exchange? A. ☐ Yes B. ☐ No

Name _____

Company _____

Address _____ City _____

State _____ Zip _____ Phone _____

Join!

A.P.P.L.E.

Apple PugetSound Program Library Exchange

THE **WORLD'S** LARGEST COMPUTER USER GROUP

Be a part of the over 16,000 A.P.P.L.E. members and you'll receive:

- 12 ISSUES OF **Call —A.P.P.L.E.** Magazine
  The nation's premier Apple computer publication.
- Substantially reduced prices on sophisticated software, hardware, publications, and services.
- Access to the A.P.P.L.E. closed user group on the **SOURCE**. Excellent for information exchange and program downloading.
- **Call —A.P.P.L.E.** Hotline.

ALL THESE BENEFITS AND MORE FOR ONLY $45!

Act Now!

☐ Please send me an A.P.P.L.E. Apple-cation and information.

☐ $45.00 payment enclosed.

☐ Mastercharge/Visa

$25.00 First time fee
20.00 First year dues
$45.00 Total dues

MasterCard # _____ Exp. Date _____

VISA # _____ Exp. Date _____

Name _____

Address _____

City _____ State _____

Zip _____ Phone _____

A.P.P.L.E., 21246 68th Ave. S., Kent, WA 98032
(206) 872-2245

First Class mail for **Call — A.P.P.L.E.** add $15. Otherwise third class mailed free as part of membership.

Additional postage required for foreign memberships and shipments.

Additional membership postage:

- Canada, West Indies, Central America, Columbia, Venezuela and Mexico — Add $15.00 ($60.00 total) **Air Mail**
- All S. American countries (except above), Europe, and N. Africa — Add $21.00 ($66.00 total)
- Asia, Mid East, S. Africa, Australia and Pacific Islands — Add $27.00 ($72.00 total)

All orders must be prepaid in U.S. funds only.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 217 CLINTON, IA

POSTAGE WILL BE PAID BY ADDRESSEE

Call —A.P.P.L.E.
P.O. Box 1993
Clinton, IA 52735

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO. 32 KENT, WA

POSTAGE WILL BE PAID BY ADDRESSEE

A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032

SOFTWARE MARKETING:

You Think You Have a Million Dollar Program

Alan D./Valerie A. Floeter

SINCE word has gotten out that we've marketed some software, we have received calls asking for information on how to do this. It seems like a lot of folks are interested in this area, so we're going to sit down and try to put our thoughts together in a little more organized fashion than some of our phone calls.

Few things in software are more satisfying than opening up a popular computer magazine and seeing an ad for something you wrote. It's a proud moment and rightfully so, considering all the effort that you have put into your package. This article will hopefully give you some background on how to get your program in the market.

Pick the Program

First off, you have to have a program to sell. For most of us this is the easiest part; our program is the one we've been sweating over every night until 1a.m. for the last four months. We know what we want to sell.

If you haven't decided that yet, think about one of these types of offerings:

1) A totally new and different concept.
2) An improvement over what's currently available.
3) An imitation of a current offering with slight changes in appearance and operation.

Let's look at each of these. Why should you duplicate a program out there? (First, let's make sure you know that we mean to write the program from scratch yourself, not copying a program for resale. That's illegal and could get you in a heap of trouble.) The number one reason to have another version is because more than one firm sells programs. If company A is selling a "whiz-bam" program, wouldn't company B also like to sell their own "whiz-bam" program, especially if orders are going like crazy. However, be prepared. If a program is selling at the top of the charts, by the time you are done with your version, it may be too late to share the gold. This is especially true in the games market.

People are always looking for the "best" program for some purpose. That way your new and improved program with added "feature-boosters" could eventually pass up the current favorite. People are interested especially in better utilities and business and educational software.

The last and best area is the totally new and revolutionary program. If you have such an idea, get on it fast, before someone else thinks of it. Companies are looking for new frontiers and if you can help them, you have a good chance of making it.

About the Authors. . .

Alan D. Floeter is currently Lead Product Engineer for Wisconsin Electrical Manufacturing in New Berlin, Wisconsin and has a BS degree in computer science.

Valerie has a BA in math and most recently was a software engineer for General Electric — Medical Division.

They have been Apple owners since 1978 and are involved in both hardware and software projects. They work under the name of The Software Experience and do accept some custom programming work for small businesses owning Apples, although they prefer to write programs for publication.

In 1979 they published their first Applesoft utility in MICRO *magazine and have since written nearly 20 articles for various computer magazines. They have been contributing editors for Nibble magazine since 1980. "So You Think You Have a Million Dollar Seller" is their first article appearing in* Call -A.P.P.L.E.

1979 also marked the introduction of their "Skybombers" game, still being sold through Instant Software. Last year Micro-SPARC started marketing "The Assembler" and "MacroSoft".

The Floeter's are currently developing a low cost printer interface for Happ Electronics and are searching for a firm to market their bubble memory system for the Apple.

Valerie and Al are making plans to present a talk on using macros in assembly language programming at the Boston Applefest in May of this year.

They have a two year old son and are expecting another child in July. They can be contacted at 4333 N. 71 Street, Milwaukee, WI 53216 (414) 463-2494.

MILLION DOLLAR PROGRAM

Do the Documentation

Once your program is finished as you see it, start working on the documentation. This is one area companies are putting more emphasis on. If you love your program, make it easy for others to get to know it and love it too. If you have a terrific program, but nobody understands what all its features are, then they aren't going to appreciate its genius. Conversely, if your program is of medium quality, but if everyone who looks at it and at your manual understands it and can find 20 ways in which to use it, they'll rate it high. Don't be afraid to say too much about your program — at least as far as your documentation goes. By the way, word processors are invaluable and, in our opinion, necessary for this phase.

Things to include in a manual are:

1) What to expect from the program. It sure is reassuring to the new user to press a key and have the screen say exactly what the manual says it should. That kind of hand-holding is great. (Remember that reviewers are often new users who aren't planning on digging out any deep dark secrets. We always like to make reviewers happy.)
2) Lots of examples — working ones. A good way of seeing how a program works is to just run through all the parts. With some ready made data supplied by the original programmer, a new buyer can see not only what must be done, but also how the writer intended the program to be used.
3) Clear description of options. A program with too many options can appear cluttered. However, if you make the distinctions between options very clear, as well as suggestions for when each should be used, you have a powerful program.
4) Reference cards. If your program has many symbols, options or keys, you may want to summarize these in a neat little page for the new user to keep on top of his/her computer. (This may work as an advertising scheme too, since the user's friends may notice the card sitting out a little more than a manual on the shelf or disk in a box.

Peer Review

Once your program and documentation is complete, hand it out to some trusty friends. Ask them for suggestions, complaints, compliments and bugs. Take their suggestions into account. Try to answer their complaints. Note their compliments; these are selling points. Fix all bugs. This is for the manual as well as the program. Remember that they are both involved in making a sale.

Market it yourself?

Up to this time we have always chosen to work with another firm in marketing our software and hardware projects. This is because we know little about marketing, don't care to learn, and would rather move onto another project once we have completed one. If you are thinking about handling it yourself, think about the following.

1) How will you advertise? A one page ad in some computer magazines is $3000 for a one shot deal. Do you have at least, say $1500 to announce your project in one magazine? What will your advertisement look like? Who will work that up for you?
2) How will you take orders? Do you have someone around full-time to answer phones? Will you hire someone? You'll need federal and state employer numbers then, and must take care of withholding and social security taxes.
3) Do you need to collect sales tax? You must file for these numbers also and submit statements quarterly.
4) Does your price take into account dealer discounts. Dealers expect somewhere around 35–50% discounts to market your program in their stores. So if you list a program at $50, a dealer will pay about $30 for it.
5) How will you package it? How will your manual look? Will you print up a nice reference card on cardboard? What sort of box, if any, will you put it in?
6) How will you handle returns, bad disks, support, technical questions, updates, fixing bugs once released, copy protection, piracy, etc.?

If you still want to do it, fantastic! You're braver than a lot of us, and should make the profit that would have gone to the marketing company.

Choosing a Marketing Company

If you do feel you'd like to let someone else worry about those details, then you need to do some studying.

1) Find out what companies are the leaders in the application you have written for and might be interested in your program. No matter if it's educational, business, utility or game, pick out about three firms that you'd like to consider.
2) Look at their current ads. How have they introduced new packages? Does your program, as well as your dreams, fit in with that?
3) What is the company's reputation? How do your local dealers view them? You can find out a lot by asking your dealer about this. How good is their support? Do a lot of dealers carry their software? If you do go with a firm, but only one out of four stores in the area carry their products, then you're not going to get the same exposure that choosing another company could get you.
4) Once you've done this research, send a letter to the firms you want to offer your program to. Tell them your application (education, business, etc), type of program (data base, PAC-man type game, program to teach sizes), type of audience you expect (children ages 4–7), system constraints (2 disk drives, RAM card, or whatever). Ask that they contact you and send a copy of their standard contract if they are interested in reviewing your program.

Reviewing the Program

Once a publisher has expressed interest in your program, the next step is to allow them to review your package. Up to this point, no contract has been signed and no commitments have been made by either of you. You can back out at any time and so can they.

Some authors we've talked to get very ugly at this stage. They become pushy, greedy, suspicious, and unreasonable. Patience, people, patience. You want to remain in good stead with your prospective publisher for many reasons.

1) You want a mutually profitable contract as soon as possible, as smoothly as possible.
2) You want them to look forward to contact with you. If they hate getting your letters and phone calls because they are so angry, they will try to put you off as long as possible, or even tell you to jump.
3) You may want to work with them when you have your second million dollar baby. If they know you as someone who works well with them, they won't be able to contain their excitement when they hear of another program by you.
4) You want them to think of you if they get an idea and need somebody to do it for them.

Authors must also think about it from the publisher's viewpoint. Would you buy a program that nobody's ever seen, written by somebody nobody's ever heard of? No, but you'd love to take a look at it if you see the promise in it. So package it up and send if off when they say they're interested. Don't expect a contract until the publisher has had ample time to consider your package, which means that they need to have it in their hands.

This is where suspicion enters in. So many authors are afraid to send object code on a disk to a publisher. They might copy it and sell it as their own, you know. Think about that for a minute.

1) If they copy it only for their own in-house use, then it must be worthy of selling, so why wouldn't they talk about an agreement with you?
2) It doesn't make sense for them to copy it and sell it as their own. We actually wish someone would do this to us. We'd hit them so fast with a multi-million dollar law suit that it would make their heads spin. After all, we have the commented source and file of correspondence. If you wanted, you could even send the package via registered letter. Not only would they lose money, but also their reputation. If word got out that company A is ripping off authors, nobody would send them any software. Where would they get their new products? No where. They'd be finished. No company is going to risk that. If they do, your court settlement will probably be more than your royalties, and you can still market your program elsewhere.

Once they have your program in hand, they'll need some time to review it. When you send the object code to them, ask how long you should wait before you expect to hear from them. Also ask to be kept up-to-date on the review process.

Expect this to take a minimum of a month. And that's of course for a straight forward, nothing special type program. During this time the publisher must be thinking about several things.

1) Does this program match up to the author's claims?
2) Does it match up to our expectations and standards?
3) How much work on the documentation will our company have to put in before we can get it into the market?
4) Is this product in line with our company's goals?
5) Is there a market for this product?
6) How many people would want a program like this? What percentage of the market can we get?
7) Based on selling projections and marketing costs, what royalty arrangement should we offer?
8) Do we want to do business with this author?

It is unreasonable to expect a marketing company to look at your program one day and return it the next. More than one person should be looking at it and after ample time to answer the above questions, they'll be in touch with you. Also remember that things such as vacations and holidays tend to slow businesses. (Thanksgiving to New Year's is an awful time for fast turn arounds. People just aren't in their offices as much.) By the way, some publishers do all the reviews in-house, others may send it to outside people who have done work for them. —◉→

☑ D.R.S.™
Can Do It!

Nothing else even comes close to D.R.S.— our Disk Recovery System—not Bag of Tricks,™ not Dark,™ not Disk Fixer™—nothing!

☑ Put your damaged disk in one drive, a blank disk in the other. D.R.S. will do what all those other guys will do...and *far more!*

☑ No other program actually reads information from damaged sectors, but D.R.S. does.

☑ No other program can read disks that were recorded off-center, but D.R.S. does.

☑ No other program can read disks with errors caused by hub damage, but D.R.S. does.

☑ No other program types, grades and fixes all possible files, but D.R.S. does.

☑ No other program makes recovered files accessible for conventional editing, but D.R.S. does.

☑ And all of this is totally automatic. D.R.S., a step ahead of the state-of-the-art in Apple disk recovery systems.

$100

SOF/SYS, INC.
4306 Upton Ave. So. • Mpls., MN 55410

Available at your dealer or call us at **(612) 929-7104**
We accept VISA/MasterCard

Circle 32 on Reader Service Card

After they take a look at it, they can say one of three things — "no, thanks," "yes, please," or "yes, but..."

If you receive a "no, thanks" then you can look at other publishers, market it yourself, or even use it as a basis for a magazine article. (Some magazines pay royalties on programs they sell on their disks.) Remember that the publisher looked at the program for at least the eight things we just mentioned, and possibly others, so one rejection doesn't mean it's a bad program. It might just mean that the company didn't want to branch off in that direction. Another company might be totally overjoyed when they receive the package.

If you got a "yes, please," then you go right to standard contracts and negotiations, but don't pass GO, nor collect $200 yet.

The "yes, but..." means changes and fixes and documentation updates, and another round of reviewing. Of course, you can jointly decide what things should be put in. Make sure you feel good about the changes you're making. It's your baby and you should be proud of it. One thing you can be assured of is that as others give suggestions, your product will be becoming smoother, better and will have a greater chance of acceptance and success in the market. Some companies start working up the contract while this refining process is occurring. Others will wait until they have a solid product.

Standard Contracts

Unless you have a huge program, as well as a new application, you will probably fit under the average, standard contract. Of course each publisher has their own "standard" contract, and there isn't one generally accepted in the field as far as we know. Contracts specify such things as:

1) Your royalty percentage. Normally something between 15–30%. We'll talk more about this later.
2) When the final product — program and manual ready for publication — is to be delivered by you to the company.
3) How soon after that point the firm is to start marketing your package.
4) If you don't want to or can't make changes or fix bugs, who pays for these changes to be done by others.
5) If the firm stops selling your program, when can you negotiate with someone else to market your program.
6) Who will pay legal fees if you or the firm goes after pirates? And who will receive any award by the court?

Generally you sign over all rights to the publisher. The publisher is the only one who will sell your program; you can't contract with anyone else to also do this. They can give it any title they want, set up advertising, assign the package price, edit the manual, and can request changes.

Negotiating a Contract

Once you've found a publisher, then comes contract time. Since most companies have standard forms, this part is usually pretty cut and dried, and handled through the mail. It is important that you've read it and discussed any parts that you feel uncertain about. That's why it's nice to have a copy of this agreement early. Call in your personal lawyer if you feel better about it. (However, he/she should know something about contracts.) If the publisher wants your product, they won't quibble on small points that would make you happier. But, if you can't come to an agreement, you'll have to start the whole cycle over with another publisher.

Royalty Payments

Once you've signed your contract and delivered your final program and manual, you can sit back and wait for the checks to roll in. There are certain things you should know about how royalties are calculated. First you receive your royalties on packages that are paid for. Some dealers order software and pay for them later, so although 100 packages have been sent out, if only 10 are paid for, you only receive royalties on those 10. Of course, eventually, you'll get your share of the other 90, but only as your publisher is paid for them.

Another thing that was confusing for us was that we didn't receive a set amount per package. For example, assume you have a $100 program, and a 25% royalty. You don't receive a flat $25 for each program sold. Since the publisher must discount your program to a dealer, that is also taken into account. You receive 25% of the money collected. So if your publisher sells your program at $60 to a dealer, you'll get $15, not the $25 that you get when the publisher sells a copy directly to a customer.

Royalty checks are usually sent out on a quarterly basis, and they are taxable. There's a line on the IRS 1040 form for royalties. You'll have to talk to an accountant if you want to claim deductions for your computer, unless you can figure it out yourself.

Now you have a head start on how to market your program. Although your experience may differ, you at least know what we've seen over the years. Good luck with your programs. We need more good software out in the market. Don't be afraid to ask a publisher if they want to look at your program. The worst that could happen is to receive a letter saying "no, thanks." All you've lost was time and postage. But if you get a "yes, please," congratulations. You've joined the swelling ranks of published authors. We need more people like you.

Programs submitted to A.P.P.L.E. follow procedures very similar to those outlined above.

Prospective software authors should send a brief outline of their program to the General Office, ATTN: Software Product Development.

HANDLING LARGE PASCAL PROGRAMS:

Getting a Quart into a Pint Pot

Chris W. Green

THE latest manuals from Apple on the Pascal System are certainly an improvement on the early preliminary manual, but, as always, there are some points that seem to have been forgotten, which result in hours, sometimes days, of wasted time in trying to implement a program. Having just completed a long journey through a large Pascal programming exercise, perhaps I can warn you of some of the potholes in the road, so that you can avoid them, or at least cross over them if necessary.

Stack Overflow.

When I started out programming in Pascal, I had thought that my problems with oversized programs were over. No more chaining, overlapping or machine language segments, I thought. A compiled language, with no lost space for the interpreter, will overcome all these problems. How wrong I was! Mind you, I should have been warned by the large section on program segmentation in the Addendum to the Apple Pascal Language Manual — it does say that some of these techniques will be crucial to success with long programs. However, I found out the hard way!

Problems with overlarge programs usually become apparent during the program operation with the appearance of the cryptic message

```
STACK OVERFLOW
Sxx, Pxx, Ixx
PRESS RESET
```

appearing on the screen. However, this message does not always appear. Sometimes, the machine just reboots without warning, or, in the Graphics Mode, presents a screenfull of garbage.

As far as I can find, there's nothing in the manual to warn you of this. Being uninitiated, I didn't even know what a "stack overflow" was, and I didn't immediately connect this abrupt message with the fact that my program had grown too large for the poor little Apple. A "trial and error" approach finally resulted in my trying again with the command mode Swapping "on" (press "S" in the command mode), which gave some degree of relief, usually only that the stack didn't overflow until later in the program!

Having diagnosed the problem, (and overcome my disappointment that my problems with large programs were not over!), I now had to find a solution (apart from just abandoning the whole exercise and hitting "H" to return to BASIC!). Here the documentation was a little more helpful, but there are still many aspects for which additional information may assist.

Include Files vs. Units.

Probably by the time your program is so large that you have experienced these problems, you will also have found that the source file of the program is so large as to be unwieldy. Although it may still fit into the Editor, it probably won't fit onto the boot disk together with its code file, even if you've removed "System.Syntax" and the error messages it provides. (Removing "System.Apple" and using the two disk boot procedure is a pain, to be avoided if at all possible.)

The obvious solution seems to be to split the source file into two or more sections, and to use the Include File option of the Compiler. Assuming you have two disk drives (you won't have progressed this far with only one, I'm sure, and like me, you probably can't afford a third, even if you wish you could!), the best approach is to put the "Include" file onto the disk "Apple2:," together with the Compiler.

However, I'm sure you'll find that the Include File option only mildly alleviates the problem, for the following reasons:

a. If the compilation fails due to an error in the Include File, hitting "E" returns you to the Editor, but with the work file, not the Include File, in the workspace. You will be given the correct error message, but the cursor will obviously be in the wrong place, since it will be in the host, not the Include File. It is therefore essential to note the position of the error in the Include File (name of procedure/function and line) and the error code before hitting "E" to return to the Editor. In fact there is no point in returning to the Editor at that stage; it is better to continue with the compilation (by hitting the space bar) and find all the errors, making a note of them as they appear, or hit "Escape" to exit. Any amendment to the Include File will then require saving the host file, getting, amending and resaving the Include File, then re-getting the host file — a lengthy process! Of course, if you're sure there'll be no errors in the Include File, then no problem — but then you certainly don't need my help!
b. No compilation time is saved. If you really have gotten the Include file correct, you don't want to re-compile it every time you make a change to the host file — compilation of a long file takes a long time, as you've no doubt found out!
c. The code file is still the same length. Thus, although you save some space on the boot disk by splitting the source file, you don't save as much as you will probably need to.

—●→

For these reasons, I abandoned the use of Include files at a very early stage and instead tried to solve the problem by separately compiling sections of the source file as Units. Here again, the manual was less than helpful, so clarification on some of the points I found difficult may assist others.

Regular vs. Intrinsic.

As indicated, there are two "flavors" of units, regular and intrinsic. Since you've almost certainly already used some intrinsic units (Turtlegraphics, Applestuff), it'll be easier to start with these.

Intrinsic units are "bound" into a "library" which effectively has to be named "System.Library" and be located on the boot disk to operate correctly. These units are not pre-linked into the program code file, but linked in automatically at run-time. This saves the time necessary to link manually after the compilation, and also has the benefit that either the host file or the unit can (to a certain extent) be modified independently of each other. Intrinsic units do, however, have some drawbacks. First, the unit must, as noted above, be bound into the System.Library. This is a rather time-consuming process and thus it is preferable that the unit code file never be changed. Second, each intrinsic unit must be allocated a code segment number, and, in some circumstances (which are not entirely clear but which are prompted by a compiler error message), a separate data segment number as well. These numbers must be less than 32, generally above 19, and also must not duplicate the segment numbers of any other intrinsic units in the library which may be used by the program. If the program already uses some of the units provided, such as Turtlegraphics, you must determine the segment numbers of these (the utility program "LIBMAP" on "Apple3:" helps here), and allocate another acceptable number to the new unit to be compiled. This number can only be changed by editing the declared number in the unit source file and then re-compiling the unit. My experience is that this complexity effectively limits the number of additional units to two or three, depending on whether they require data segments.

For these reasons, intrinsic units are best limited to long general purpose routines which may be used by several programs, and which have been thoroughly and completely debugged. Examples I have used are a set of procedures and functions for "bomb-proof" input of reals and integers, a graphics printer driver and a unit to read disk directories and display the file names. Intrinsic units are best not used for normal segments of a program which are likely to require frequent modification. For this purpose, the regular unit is much more appropriate.

AMPERWARE ???

You're Kidding? Not Another Amper-Utility!

Yes folks, another amper-utility; but not *just* another amper-utility. Amperware is a very special extension to Applesoft* which can make your application program rival the best commercially available. You get all the power of assembly language without worrying about registers, flags, stacks and similar nonsense. Just add three lines to your HELLO program and Applesoft* is immediately extended to include many new commands like:

& INPUT - create powerful video forms
& FIND - fast string array searching
& SORT - rapid array sorts
& READ/WRITE - quick disk access in binary (up to 16 x faster!)
& PRINT WITH - complete formatted printing

Amperware was written with the novice programmer in mind. No PEEKs, No POKEs, No CALLs . . . anyone can use it. The assembly programmer will find it a good starting point which can be extended as needed. Whatever your skill level, we think you'll like Amperware. At this price you can't go wrong.

ONLY $49.95** (+ $1.50 shipping)

(Indiana Residents Add 5% Sales Tax)

Ask at your local dealer or order directly from:

Scientific Software Products, Inc.

3171 Donald Avenue
Indianapolis, In 46224
(317) 299-0467

*Applesoft is a registered trademark of Apple Computer Inc.
**Master Card and VISA accepted.

SSP inc.

Circle 30 on Reader Service Card

The regular unit is compiled in the same way as an intrinsic unit, except that no segment numbers are required. There are two options available for making use of them. First, they can be bound into a "library" file, using the "Apple3:Library" utility. This library file must be located on the boot disk with the name "System. Library" at program compile time. If intrinsic units are used, they must also be bound into the same library, since the Compiler will also need to access these during compilation. With this method, the units can be called in the host program using "Uses ..." in the same way as intrinsic units, and, at link time, the Linker automatically finds the units in the library. However, the need to bind the units into a library and move this library around appears to make this a cumbersome and difficult process, gaining little over intrinsic units for convenience.

The second method, which I have found to be much better, is to compile the units in the normal way, then transfer the code file to "Apple2:." The host file, and any other unit which makes use of routines in this unit, will then require the following declaration:

"Uses (*$U Apple2:UGubbins *) Gubbins"

where "UGubbins" is the name of the code file, and "Gubbins" is the unit name. If, for example, the program is broken into a host and four units, then the host file will have four file identifications after the "uses," in addition to any intrinsic units which are also used. Note that the Compiler requires access to "Used" units so they, or the library into which they're bound, must be available in one of the drives throughout compile time.

In addition to putting the unit code files onto "Apple2:," I have found it preferable to also transfer the unlinked host codefile to "Apple2:." With "Apple2:" designated as the Prefix Volume, the linking process is then fairly simple. After calling the Linker, the host file name is typed in response to the request for its name, and each unit file name is then entered in response to the prompt for library files. I generally hit return when asked for the output file name, which causes the linked code file to be placed in System.Wrk.Code on the boot disk; this can now be Run without further linking. If the code file is likely to be long, it is preferable to clear the workfile with the New command from the Filer before linking, to ensure that space is available to store the linked codefile — the Linker (like the Compiler) doesn't find out that there's not enough room until it has reached the end of its task!

Unfortunately, when using this method, the Apple cannot "read" the host file "Uses" declaration to determine the names of the library files to be linked, and these must therefore be typed in one after the other. To me, this is unforgivable and hopefully a later version of the Linker will be able to carry out this task without assistance.

Following linking, the linked code file can be "saved" using the Filer. It is preferable to save it under a different name than the unlinked host code file, since this unlinked file may be needed for a re-link if any of the units it uses are subsequently changed.

If a unit is modified, all that is required (assuming it does not affect any of the other units or the host) is for the re-compiled codefile of the unit to be re-saved to "Apple2:," and the whole re-linked. Note that despite the note on page 81 of the language manual, the host program need only be re-compiled if changes are made to the public part of the unit which it uses. Therefore any changes in the interface part of the unit will mean that any other files which use the unit will require recompilation. This is also true of intrinsic units, and this makes it doubly important that general use units are thoroughly debugged before they are added to the library.

I have found that the easiest way to go through the process is to "Get" the file to be worked on, edit and compile it, then, after entering the Filer, "Transfer" "*System.Wrk. Code" to "Apple2:" with its correct name. I then "Save" the workfile back to its original disk and name. Note that it is not necessary to have the suffix ".CODE" on the name of the file on "Apple2:" — the Linker does not require it. However, the "Uses" declarations must obviously reflect the actual names used.

Use of Files in Units.

As noted on page 79 of the language manual, variables of type "file" declared in the implementation part of a unit will cause a "No Private Files" compiler error to be flagged. This is a serious deficiency, and it complicates the control of printer or disk from inside a unit. For example, if you require outputing to a printer with the statement

"Rewrite(Fi, "Printer:");"

the declaration of the variable "Fi" as a file of type interactive cannot occur inside the procedure.

There are two possible solutions which I have found to this problem. First, the file variable declaration can be placed in the "Public" or interface part of the unit. This works, but requires the use of a separate data segment if the unit is intrinsic, and for some reason seems to reserve a huge amount of memory space in the program, even when the unit is non-resident. This took me a long time and many a "STACK OVERFLOW" message to discover, since the reserved memory space can be as much as one third of the available memory!

A second and better alternative is to pass the file identification to the procedure with a value parameter, e.g.:

"Procedure PrintText(Fi : Interactive);"

and call this from a host file routine which declares and defines the variable file to be passed. This seems to work without problems especially for Text and Interactive files, but is a bit clumsy for files of data.

Removing Units from System.Library.

As you add intrinsic units to "System.Library," the library file of necessity becomes larger and once again restricts the use of "Apple1:." You will then probably be tempted to remove from the library any units which are not required in an attempt to alleviate the problem.

GETTING A QUART INTO A PINT

There appears to be no problem in removing normal intrinsic units — Applestuff, Turtlegraphics, Chainstuff or Transcend — as long as none of the programs which might use the library requires them. There are two undocumented units in the library — "LONGINTI" and "PASCALIO" — and you may be tempted to remove these, especially if you are not using long integers or the procedure "Seek." Unfortunately, I found that removing these units gave rise to problems at boot time when using any file named "System.Startup", even though the startup file doesn't appear to use either unit. At best, the message "Required Intrinsics not Available" appeared. At worst, the system seemed to die. It does appear, therefore, that these two units *must* be left in the System.Library at all times.

One final small point on the linking process. I have noted that if any copyright or other notices are embedded in the host code file using the (*$C...*) compiler option, these notices get lost during the linking process. There is therefore little object in including such notices in files which will subsequently be linked. If such notices are essential, they must be embedded using a PascalZap type of program.

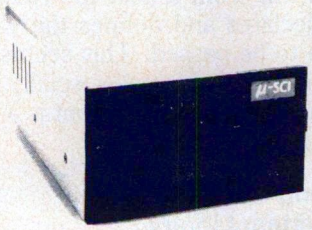
Library Notices.

The "Library" utility on "Apple3:," which allows you to bind units together into new libraries, also allows you to append a notice to the library. The example in the manual indicates that this can be used to declare the copyright. While this may (or more probably, may not!) be useful, I have found it essential to use this notice to identify the library. You will find that you will bind together special libraries for tasks specific to the programs which will run with them. These will (must) all end up with the name "System.Library" if they are to be used, and one can get very confused about what is in each one. I have therefore identified my libraries as "Library.1", "Library.2" and so on, and this identification, together with the binding date, is what I enter at "notice" time. Then, if I am in any doubt as to which library is on the disk, a quick run of "Apple3:Libmap" will give an immediate answer.

Program Segmentation.

Breaking the program down into separately compiled and linked units certainly helps to make the source file easier to handle, and the compiling time is much reduced. It does not, however, immediately solve all the problems of lack of memory space to RUN the program, although it does provide the wherewithal to attack this problem.

Laing Electronics



μ-SCI

MICRO-SCI

A2 DRIVE

$235

BUY THE FIRST AND THE BEST APPLE COMPATIBLE DRIVE!

A2 DRIVE WITH CONTROLLER	$306
A40 (40 TRACK DRIVE W/O CONTROLLER)	$284
A70 (70 TRACK DRIVE W/O CONTROLLER)	$365
-CONTROLLER FOR A40 OR A70 DRIVES	$ 79
A3 APPLE III DIRECT REPLACEMENT	$299

10382 Stanford Ave., E-203, Garden Grove, Cal., 92640

Phone (714) 534-4216

Check or Money Order, - NO CREDIT CARDS - Check's Allow 10 BANKING DAYS FOR CLEARANCE
Prices and specifications subject to change without notice.
Add $7.50 for UPS surface. Cal. Res. add 6% Sales Tax.

Circle 11 on Reader Service Card

HOT PRICES!

SYSCOM COMPUTER
48K $679
128 K ram card, controller, 80 col. card . . . . CALL

MONITORS

BMC 12" green	$85
Amdek amber	$159
Taxan RGB 1 w/interface card, cable	$408

DISK DRIVES

Davong 5mb hard disk	$1529
Shugard 35 track w/half tracking	$239
Rana Elite One	$289
Half Height drives	$219

PRINTERS

Gemini 10, 100cps, 2.5K buffer (Epson clone)	$359
Gemini 15, wide paper capability	$515
All other brands	CALL

PERIPHERALS

Viewmax 80 column card	$199
Grappler Plus	$139
Hayes Smartmodem w/interface, software	$269
For Other Products PLEASE	CALL

PACIFIC COMPUTERS

ORDERS ONLY 800-531-3133 • INFORMATION (206) 641-7233
13256 Northup Way #7, Bellevue, WA 98005
Prices reflect 3% cash discount. Bank Cards

Circle 25 on Reader Service Card

Each unit is treated as a separate program segment, whose number is either allocated at compile time (for intrinsic units) or at link time (for regular units). However, under normal circumstances, all segments are read into memory at the time of execution, and if the result overflows the memory, the result is predictable if you've been following this far — yes, a "STACK OVERFLOW" message! This can be overcome by inserting the "Non-Resident" compiler option "(*$N+*)" as the first line of the main section of the program. Each segment is then only read into memory when it is required, and swapped out again when it is finished with.

The Addendum to the language manual gives relatively clear examples of the use of this option, together with the use of the associated "Resident" option, which will retain the segment in memory for a longer period, in cases where the unit or segment is called repeatedly, as in a loop. What may not be so clear is that the action of loading the segment at any stage may be the "last straw" to the memory, and result in a stack overflow. If this occurs while in the graphics mode, the message may not be seen, and the system just dies with garbage on the screen. Since "Turtlegraphics" is a very large segment, requiring almost half the available memory, the crash quite often occurs when an attempt is made to use graphics commands. Note also that Turtlegraphics seems to reserve 7183 words of memory space even when non-resident. I assume that this protects the High-Resolution graphics page 1 from overwriting — whereas in BASIC, the programmer always needs to be concerned about overwriting the graphics page, this need not concern the Pascal programmer.

Another factor with segmentation and the "Non-Resident" option is that the disks spend much of their time grunting and whirring throughout the program operation. Thankfully the access is so much faster than DOS, that the delays this introduces are hardly noticeable, and this certainly does not seem to give rise to problems other than those mentioned above. Using the compiler "Resident" option can, of course, reduce this excessive disk activity.

It is not possible to segment units, but with careful programming, this is not essential. However, if procedures which require large amounts of memory (particularly those with many string constants, such as menus or instructions) are kept in the host files, they can then be separately segmented and reside in memory only when required. Remember that no more than 16 segments are allowed, including the main program, thus the total of regular units plus segmented procedures and functions in the host may not exceed 15. Counting intrinsic units, a maximum of 26 segments is allowed. More than 10 intrinsic units are allowed, but only at the expense of regular units or segment procedures/functions.

Since the segment table for regular units and segment procedures/functions starts at number 7, using the maximum of 15 such would result in segments numbered up to 21. Unfortunately, the two "Turtlegraphics" segments are numbered 20 and 21; if graphics are required, these numbers cannot be used for program segments. As it is not possible to change the segment numbers of intrinsic units unless the source code is available, a means to avoid this clash is required. Happily, Apple has provided one, in the form of the "Next Segment" compiler option "(*$NS num*)". This option command must be included immediately before the segment which would otherwise be numbered 20. The "num" must refer to an unused unit segment number, and there must be contiguous unused segment numbers for all the subsequent segments, if any, or another "Next Segment" option will be needed. Note that these numbers may duplicate existing unit segment numbers as long as the units with these segment numbers are not "used" by the program. These numbers will depend on which segment numbers have been allocated in the user's library unit, and which original units have been removed. Unfortunately, once again the computer cannot handle this relatively simple task of allocating segment numbers to avoid a clash, and so it becomes the programmer's responsibility to ensure that the "Next Segment" option is included where necessary, together with an acceptable segment number. In counting up, bear in mind that the regular units are, first, allocated numbers, starting at 7, followed by the segments in the host program in the order in which the main body of the segment procedure/function appears in the program.

It must be noted that the segment procedures/functions must appear as the first procedures/functions in the host program. Thus, if a short, general use procedure is required which will be called by segment procedures or functions, and therefore requiring it to be the first procedure in the file, it must normally be itself a segment. If, due to limitations on the number of segments, you do not wish to make this a segment, the solution is to declare all the segment procedures/functions at the beginning of the program using a "Forward" declaration. The remainder of the program can then be written as normal.

Nested segments are allowable, thus the following will work:

```
Program NitWit;

  Segment Procedure Nincompoop;

    Segment Function Twit : Real;

      Begin
        ...
      End; (* Twit *)

      Procedure BirdBrain;

      Begin
        ...
      End; (* BirdBrain *)

    Begin (* Nincompoop *)
      ...
    End (* Nincompoop *)

    Segment Procedure FanTong;

    etc, etc...
```

You will see that, although "BirdBrain" is not a segment, "FanTong" which follows may still be a segment. However, if "BirdBrain" appeared before "Twit" in "Nincompoop," then "BirdBrain" would need to be a segment.

Using these various tools, it is possible to segment the program, such that only the larger procedures/functions are segmented, while the short ones remain unsegmented.

GETTING A QUART INTO A PINT

Chaining Programs.

A more extreme approach which may be necessary for very large programs is to use the "Chain" procedure, again covered reasonably adequately in the Addendum to the language manual. As noted there, data can be passed between programs by using disk files. If this approach is used, it is preferable to "Make" the required file on the boot disk before the program is run and leave it there. This will avoid the possibility of the chain failing due to lack of disk space.

One further point on chaining — it appears to be impossible to chain back to "System.Startup," therefore if a turnkey disk is planned with "System.Startup" providing the main menu, it can only be used for the boot situation. Exiting to the main menu requires chaining to another file name. There are two possible solutions - either make System.Startup an almost dummy program which does nothing but chain to the main menu program, or duplicate the System.Startup program on the disk with another name which can be used for the chain command.

Other General Points.

Although not directly connected with the problems of long programs, there are a couple of other general points which are related. The first returns to the troublesome "STACK OVERFLOW" problem, which can also occur during compile time. To alleviate this problem, the compiler and possibly the operating system must be placed into "swapping" modes. The compiler has two swapping modes (*$$+*) and (*$$++*) and the operating system has one — set by the S command. As a general rule, if the space available drops below 550 words (those numbers displayed in brackets during compilation) you will need to employ some kind of swapping. This will lengthen the time spent compiling but it is better than the "STACK OVERFLOW" condition. I have found that if I wish to interrupt a compile operation, I might as well use the Reset key, since otherwise I'll certainly have to Reset later. A user break (control @) looses track of memory and will eventually cause a "STACK OVERFLOW". Perhaps this can be overcome in a later issue of the system.

The other shortcoming I have found with Pascal 1.1 is that it is not possible to retain a high-resolution display on the screen while chaining into another program. This may seem a minor drawback (I guess it probably is!) but it would be nice in programs using graphics to fill the screen with an appropriate title page or notice while the main program is loading - loading a program with many intrinsics can take some time, with much disk grunting!

Summary.

Pascal does not immediately solve all the problems of large programs which occur in BASIC, but use of units (both regular and intrinsic) together with sensible use of segments conditioned by the No-Load and Resident options in the Compiler can allow quite large programs to be accommodated. This may cause considerable disk activity while the program is running; consequently, it demands that the disk containing the program (and probably System. Library) remain in the same drive throughout its operation.

As with many aspects of the Pascal system, the use of these techniques takes some getting used to, but once this has been done, the full flexibility of the system becomes apparent. I, for one, now much prefer to use Pascal compared with the BASIC I first learned.

BEEP! LOST YOUR PROGRAM?

BUS RIDER
LOGIC ANALYZER FOR THE APPLE II

The Bus Rider is a self diagnostic development tool that allows real time analysis of software and hardware in the Apple II computer.

The Bus Rider provides:

- Monitors and saves 512 cycles of the address and data bus, NMI, IRQ, DMA, R/W and 4 external lines.
- Pretrigger viewing of up to 512 samples.
- 4 external inputs with variable threshold reference.
- Display cycle by cycle execution or 6502 disassembled code.

The Bus Rider comes complete with Bus Rider circuit card, reference manual, Bus Rider software diskette, and 10 easy hook external input cable. The total system price is **$395.00**

Bus Rider - Disassembled Display

RC Electronics Inc.
5386 Hollister Avenue, #D
Santa Barbara, CA 93111
(805) 968-6614 • TELEX 295281

MC Visa

Circle 29 on Reader Service Card

A **Call -A.P.P.L.E.** APPLE-CATION:

Magfiler

Jack Olivieri

Introduction

THE information currently available about the Apple Computer (both hardware and software) is massive. No one can hope to remember which article showed the Apple user how to check RAM memory automatically or how to make the six color modification. This explosion of information occurs in many fields of knowledge and gets worse each day. The information is there, but it is difficult to find, and many hours can be spent doing a literature search.

For this reason, I wanted a simple (single string), personal periodical guide to the magazines that I subscribe to. I wanted it to store article titles, magazine name and issue number. It should be expandable to more than one disk drive and allow at least an alphabetical sort of the article title over the entire data base, and it must be simple enough to adapt to other applications such as a video tape, music or book libraries. Of course, it must have a search capability. As a result of these requirements, MAGFILER was developed using a presorting, multifile technique, capable of 2000+ records.

How MAGFILER Works

The first and most serious problem when manipulating a large data base is the amount of RAM memory available for record storage. This is a fixed quantity. In most simple programs this will limit the number of records because sorting the file will usually require that each record be compared to each other record. It is also desirable to have the entire data base reside in memory from a speed point of view. If no excursions are made to and from the disk, then a lot of time is saved. However, there is that record limitation.

One alternative would be to access individual records as required from the disk to do the sort. However, this would quickly wear out the drive and take forever (almost) to sort.

What MAGFILER does is a compromise to increase the amount of records the Apple can sort. When a record is first entered, the program will look at the first character and based on the ASCII value, assign that record to one of eight files. In this way the program "presorts" the information as it is entered into files which are small enough to fit into the RAM memory of the Apple. If each file is filled to capacity, then this will increase the number of records we can handle by eight. When a sort is performed, each file is read from the disk. You are limited to 16 excursions to and from the disk for the entire data base. All other functions of adding, printing, editing, searching and deleting take into account this multifile configuration. Now let's look at these operations in more detail.

File Structure

All files are identical. They are random access files with 65 character records. The first record (record 0) in each file contains a number, indicating how many records are in the file. The file name for the data base is entered in lines 0 and 1 (NA$).

Initialization (Lines 1 & 270-360)

The program is initialized by typing "RUN 1." This will either a) set up new files on a clean diskette or b) wipe everything out of the files already on the diskette and set up empty files. The program will take the name of the data base files (MAGFILE) and add to it a number from 1 to 8 and setup the eight files called MAGFILE1,MAGFILE2,MAGFILE3,...etc. Also a zero will be placed in record 0 of each file to indicate the number of records in the file.

Running the Program

In normal use the program should be "RUN." This will bypass line 1, cause certain variables to be defined and a READ to be performed on record 0 of each text file to determine the number of records in each file and total number of records (lines 90–145).

Menu (Lines 160-260)

A standard menu will be displayed asking you to choose one of the following:

1. ENTER DATA (LINES 370–600)

The entry routine makes it easy to enter data from the keyboard. The program prompts with the words "TITLE:," "MAGAZINE:," and "MO./YR.". The title is limited to 40 characters, magazine name to 10 and the mo./yr. (or issue no.) to 10 characters. If any of these are exceeded, the program will only take the first 40 characters from the first field and the first 10 from the second and third fields. In case you prefer not to count characters, a ":" will be displayed to show where the end of that entry is located. To prevent a lot of work, a copy feature is provided so that when entering articles from the same magazine, the magazine name and mo./yr. need only be entered once. After that, simply hit "RETURN" for the second and third entries and they will be the same as the last typed entry. The final string is concatenated from these smaller strings (A$, B$, C$) in line 470 and right and left arrows are added to set the title apart for readability. The entries are sent to the disk after each ten entries to protect against a program or power crash. Also the records are sent to the disk when a [RETURN] is typed for the article title, which indicates that you are finished entering data.

—◉→

The short subroutine (lines 474–479) sets up an index list which indicates which records go to which file.

When the number of records in a particular file approaches 300, a message will be printed indicating which file is becoming filled up.

2. PRINT ALL RECORDS (LINES 740–1140)

When the printing option is chosen then the program will ask if hardcopy is desired. If the answer is "NO," "N" or just [RETURN], then the output will be formatted to the screen. This will indent the record number (POKEs in line 930) and check for the end of the page (line 952) and print an "end of page" message (line 1000). At this point hitting [RETURN] will continue the listing and "esc" will return you to the menu.

The print routine reads a file at a time starting at MAGFILE1. The entire file is read at once and then printed a page at a time until all the records are exhausted. Then the next file is read. If there are not enough records in a particular file to fill the page, the next file is read and so on. After all the files have been read and all the records printed, then an end of records (line 1120) message is printed.

The read records loop (line 830) is contained within the open file loop (line 782). NU is the number of files and M%(I) is the number of records in file MAGFILE.

3. SEARCH (LINES 900 & 910)

The search option is contained within the printing part of the program. When called, the program will ask "KEYWORDS LOOKING FOR ". The word or phrase that you are searching for should now be entered. Every record will be printed which has the word or phrase imbedded anywhere within it. When entering data, remember to type in a word the same way each time to represent the same thing. For example, I always try to type "HI-RES" for everything that has to do with the high resolution capability of the Apple. In this way, the search routine will capture and print everything, which it would not do if I typed "high res", occasionally.

Also the search routine can be used to perform a quasi-sort. To get all the magazines of the same title, or same year, simply type the correct keyword(s) in and all records with the same magazine title, year or even month will be printed.

4. CHANGE A LINE (LINES 1150–1320)

To change (edit) a line, type "4" and enter the number of the record that must be changed. The record indicated will be displayed with the flashing cursor over the first character of the record. If the record is correct, as is, then hit [RETURN] and the record will stay the same. To edit the record use the right arrow (→) to copy over the correct characters and then make the changes. Remember to copy the entire record before hitting [RETURN]. An "*" will appear at the end of the maximum record length (65 characters).

The change routine first uses a subroutine (lines 1174–1220) to find in which file the record is resident and where in the file the record is located. (While the "PRINT ALL RECORDS" listing starts at one and numbers the records consecutively, each file has M%(I) records that are numbered from one also.) This subroutine also displays the record for verification.

Next the new record is stored in A$ and the old record is deleted from its file, the file is renumbered and the record count for that file is decremented as well as the total record count. Then the new record is analyzed (lines 474–479) to see in which file it belongs and lines 1250–1286 put it into the file as well as incrementing the proper record counting indexes (M%(I) & M).

5. DELETE A LINE (LINES 1401–1427)

The delete routine does half of what the "CHANGE A LINE" option does. The same subroutines are called

```
0 NU = 8: PRINT  CHR$ (4);"MAXFIL
     ES ";NU:D$ =  CHR$ (4):NA$ =
     "MAGFILE":Z4 =  PEEK (54):Z5
      =  PEEK (55):Z6 =  PEEK (56
     ):Z7 =  PEEK (57): GOTO 90
1 NU = 8: PRINT  CHR$ (4);"MAXFIL
     ES ";NU:D$ =  CHR$ (4):NA$ =
     "MAGFILE": GOTO 280
2  IF Z4 = 240 THEN  RETURN
3  POKE 54,240: POKE 55,253: POKE
     56,27: POKE 57,253: RETURN
4  POKE 54,Z4: POKE 55,Z5: POKE 5
     6,Z6: POKE 57,Z7: RETURN
5  POKE 43624,1: RETURN
10  REM

**********************************
* MAGFILER * BY  JACK OLIVIERI *

* CALL -A.P.P.L.E. * MAY, 1983 *
**********************************

90 SZ = 302:M = 0:SE = 10:GG$ =  CHR$
     (7) +  CHR$ (7) +  CHR$ (7)
95 B%(1) = 65:B%(2) = 68:B%(3) =
     71:B%(4) = 75:B%(5) = 79:B%(
     6) = 83:B%(7) = 87:B%(8) = 9
     5
```

```
100  DIM A$(SZ),A%(SE + 1): GOSUB
     4: FOR I = 1 TO NU: GOSUB 5
110  PRINT D$;"OPEN";NA$;I;",L65"

120  PRINT D$;"READ";NA$;I;",R0"
130  INPUT M%(I)
135  NEXT I
140  GOSUB 2
145  FOR I = 1 TO NU:M = M%(I) +
     M: NEXT I
150  REM
**********************************
160  REM
*** PRINT MENU ******************

170  HOME : PRINT  TAB( 14): PRINT
     "M E N U "
180  PRINT : PRINT "1. ENTER DATA
     ": PRINT : PRINT "2. PRINT A
     LL RECORDS": PRINT : PRINT "
     3. SEARCH"
190  PRINT : PRINT "4. CHANGE A R
     ECORD": PRINT : PRINT "5. DE
     LETE A RECORD"
200  PRINT : PRINT "6. NO. OF REC
     ORDS"
210  PRINT : PRINT "7. SORT"
211  PRINT : PRINT "8. EXIT"
220  PRINT : PRINT
```

to find and display the proper record for verification (1174–1220) and then renumber and write the file back to the disk (1402–1427). The program will allow an abort back to the menu before doing the actual deletion.

6. NUMBER OF RECORDS (LINES 1598–1610)

By typing "6," the number of records in each file will be displayed along with the "boundaries" or ASCII values for each file that the program uses to determine what record goes where.

7. SORT (LINES 1430–1596)

The sort is a modified shell sort which sorts approximately 100 records per minute. When sorting, the file that is being sorted will be displayed. Any files which have less than two records do not need sorting and will be skipped over. After each file is sorted, it is written back to the disk in sorted order. All records are sorted in ascending (ASCII) order alphabetically and the actual sorting subroutine is located at lines 1470–1595.

8. EXIT (LINE 1620)

The "EXIT" selection should be chosen whenever all work is completed to insure that all open files will be closed.

Subroutines

A table of subroutines is listed in TABLE 1. In addition to the ones already mentioned, there are other ones which have many applications in other programs. The subroutine at lines 2–3 turns DOS off so that input from the keyboard and output to the screen will not be picked up by DOS. The sub at line 4 will turn DOS back on when needed for disk operations. That is why, if you break out of this program without using the "8.EXIT" selection on the menu, you could end up without DOS. If a CATALOG command is attempted for instance, a "SYNTAX ERROR" will result.

Another useful subroutine is the one at lines 5–6.

This selects the drive number without having to "CATALOG" or issue some other DOS command first. For drive 2 simply POKE 43624,2. (POKE 43626,x where x is the slot no. is also handy).

TABLE 1

Subroutine	Lines
Turn DOS off	2-3
Turn DOS on	4
Set Drive Number	5-6
Assign File #	474-479
Find Correct File, Record, & Print	1174-1224
Delete Record, Renumber File	1402-1426
Sort	1470-1595
Find # of Records in Each File	1600-1607

Modifications

The number of input strings can be changed from 3 to any other number, as long as they are concatenated to make up the final string (A$(I) in line 470) and it is less than 65 characters long. This record size could be increased, of course, in every OPEN statement, but would require a subsequent reduction in the number of records that could be handled at once.

```
230  PRINT "CHOOSE ONE : ";: INPUT
     V5
240  ON V5 GOTO 380,760,760,1160,
     1330,1600,1430,1620
250  GOTO 170
260  REM
**********************************
270  REM
*** CREATE NEW FILES MODULE ****

280  REM
285  FOR I = 1 TO NU: GOSUB 5
290  PRINT D$;"OPEN";NA$;I;",L65"

300  PRINT D$;"DELETE";NA$;I
310  PRINT D$;"OPEN";NA$;I;",L65"

320  PRINT D$;"WRITE";NA$;I;",R0"

330  PRINT 0
340  NEXT I
350  PRINT D$;"CLOSE": END
360  REM
**********************************
370  REM
*** ENTER NEW DATA MODULE ******

380  HOME
390 EN = 0: FOR I = 1 TO SE: PRINT
     : PRINT  SPC( 14);":": VTAB
      PEEK (37) - 1: PRINT M + I;
     ". ";: HTAB 7: PRINT "TITLE:
      ";: INPUT A$: IF A$ = "" THEN
     EN = 1: GOTO 620
400 L$ = B$:M$ = C$
410  HTAB 14: PRINT "     :";: HTAB
     1: PRINT "MAGAZ:";: INPUT B$
     : HTAB 19: PRINT ":";: HTAB
     1: PRINT "MO./YR:";: INPUT C
     $
420  IF  LEN (A$) > 40 THEN A$ =
      LEFT$ (A$,40)
430  IF  LEN (B$) > 10 THEN B$ =
      LEFT$ (B$,10)
440  IF  LEN (C$) > 10 THEN C$ =
      LEFT$ (C$,10)
450  IF B$ = "" THEN B$ = L$
460  IF C$ = "" THEN C$ = M$
470 A$(I) = A$ + " (" + B$ + "-" +
     C$ + ")": GOSUB 474: NEXT I:
      GOTO 620
474 A%(I) =  ASC (A$)
476  FOR K = 1 TO NU
477  IF A%(I) <  = B%(K) THEN A%(
     I) = K: RETURN
478  IF (A%(I) > B%(K)) AND (A%(I
     ) <  = B%(K + 1)) THEN A%(I)
      = K + 1: RETURN
479  NEXT K: PRINT "CAN'T FIND CO
     RRECT FILE FOR ": PRINT : PRINT
     : PRINT A$(I): PRINT : PRINT
     "CHECK FIRST CHARACTER": END

600  REM
**********************************
610  REM
*** WRITE RECORDS TO DISK MODULE

620 P = I: IF I = 1 THEN 170
622  GOSUB 4: FOR I = 1 TO NU: GOSUB
     5
624  PRINT D$;"OPEN";NA$;I;",L65"

626  NEXT : FOR J = 1 TO P - 1:I =
     A%(J): GOSUB 5
650  PRINT D$;"WRITE";NA$;I;",R";
     M%(I) + 1
660  PRINT A$(J):M%(I) = M%(I) +
     1: NEXT
685  FOR I = 1 TO NU: GOSUB 5
690  PRINT D$;"WRITE";NA$;I;",R0"
```

```
700  PRINT M%(I)
705  NEXT : PRINT D$;"CLOSE"
710  GOSUB 2
720 M = 0: FOR I = 1 TO NU:M = M +
     M%(I): IF M%(I) )  = SZ - 15
      THEN  PRINT : PRINT : PRINT
     NA$;I;" IS GETTING FILLED UP
      ";GG$: FOR J = 1 TO 1500: NEXT
     J
722  NEXT I: IF EN = 1 THEN 170
725  GOTO 390
730  REM
*********************************
740  REM
*** PRINT ALL RECORDS IF V5=2 **

750  REM
*** SEARCH & PRINT     IF V5=3 **

760  HOME
770  IF V5 = 3 THEN  INPUT "KEYWO
     RD(S) LOOKING FOR  ";CL$: PRINT

780  INPUT "WANT HARD COPY (Y/N)
     ";H$: HOME
782 B = 1: FOR I = 1 TO NU: GOSUB
     4: IF M%(I) = 0 THEN  NEXT I
     : GOSUB 2: GOTO 1120
815  GOSUB 5
820  PRINT D$;"OPEN";NA$;I;",L65"

830  FOR J = 1 TO M%(I)
840  PRINT D$;"READ";NA$;I;",R";J

850  INPUT A$(J)
860  NEXT J: GOSUB 2
880  IF  LEFT$ (H$,1) = "Y" THEN
      PR# 1
890  FOR J = 1 TO M%(I): IF V5 =
     2 THEN 930
900 L7 =  LEN (A$(J)):C8 =  LEN (
     CL$): FOR K7 = 1 TO (L7 - C8
      + 1): IF  MID$ (A$(J),K7,C8
     ) = CL$ THEN 930
910  NEXT K7
920 B = B + 1: NEXT J: NEXT I: GOTO
     1070
930  POKE 33,40: POKE 32,0: PRINT
     : PRINT B;"  ";: POKE 33,36:
      POKE 32,4: PRINT A$(J):B =
     B + 1
935  IF  LEFT$ (H$,1) = "Y" THEN
      NEXT J: NEXT I: GOTO 1070
952  IF  PEEK (37) =  ) 18 THEN 1
     000
960  NEXT J: NEXT I: GOTO 1070
1000  PR# 0: PRINT : INVERSE : PRINT
     "'RTN'= MORE  'ESC'=STOP": NORMAL

1010  GET S$
1020  IF  ASC (S$) = 13 THEN  POKE
     33,40: POKE 32,0: PRINT : HOME
     : NEXT J: NEXT I: GOTO 1070
1030  IF  ASC (S$) = 27 THEN  PR#
     0: POKE 33,40: POKE 32,0: PRINT
     : HOME : GOTO 170
1040  GOTO 1010
1070  GOSUB 2
1120  PR# 0: PRINT : PRINT "THAT'
     S ALL FOLKS !!!": PRINT "PRE
     SS RETURN TO GET BACK TO THE
      MENU "
1130  GET G$: POKE 33,40: POKE 32
     ,0: HOME : GOTO 170

1140  REM
*********************************
1150  REM
*** CHANGE ENTRY MODULE ********

1160  HOME : PRINT "ENTER NUMBER
     OF ENTRY TO BE CHANGED ": INPUT
     N1:
1162  IF N1 = 0 THEN  GOSUB 2: GOTO
     170
1170  IF N1 ) M THEN  PRINT : PRINT
     : PRINT "RECORD DOES NOT EXI
     ST ": FOR JH = 1 TO 1000: NEXT
     JH: GOTO 1160
1172  GOSUB 1174: GOTO 1230
1173  REM
*** SUB TO FIND RECORD,
    FILE AND PRINT IT **********

1174 MB = 0:MT = 0:N2 = N1
1175  FOR I = 1 TO NU:MT = MT + M
     %(I): IF N1 (  = MT THEN N2 =
     N2 - MB: GOSUB 4: GOTO 1180
1177 MB = MB + M%(I): NEXT I
1180  GOSUB 5: PRINT D$;"OPEN";NA
     $;I;",L65"
1190  PRINT D$;"READ";NA$;I;",R";
     N2
1200  INPUT Z$
1210  PRINT D$;"CLOSE"
1220  PRINT : PRINT "ITEM # ";N1;
     " CURRENTLY IS ": PRINT : PRINT
     " ";Z$: RETURN
1224  REM
*********************************
1230  HTAB 26: VTAB 7: PRINT "*(E
     ND)": PRINT : PRINT "ENTER R
     EVISED DATA NOW ": PRINT
1240  VTAB 6: HTAB 1: INPUT A$: IF
     A$ = "" THEN A$ = Z$
1245  GOSUB 1402:I = 0: GOSUB 474
     : GOSUB 1250:M = M + 1: GOTO
     1290
1250 I = A%(0)
1252  GOSUB 5: PRINT D$;"OPEN";NA
     $;I;",L65"
1260  PRINT D$;"WRITE";NA$;I;",R"
     ;M%(I) + 1
1270  PRINT A$
1280  PRINT D$;"WRITE";NA$;I;",R0
     "
1282 M%(I) = M%(I) + 1
1284  PRINT M%(I)
1285  RETURN
1286  PRINT D$;"CLOSE"
1290  GOSUB 2: VTAB 12: PRINT "MO
     RE CHANGES (Y/N)": INPUT P$
1300  IF  LEFT$ (P$,1) = "Y" THEN
     1160
1310  GOTO 170
1320  REM
*********************************
1330  REM
*** DELETE MODULE **************

1340  CALL  - 936: PRINT "ENTER N
     UMBER OF ENTRY TO BE DELETED
      ": INPUT N1
1350  GOSUB 1174
1380  PRINT : PRINT "ARE YOU SURE
      YOU WANT TO DELETE  THIS ??
     ?": INPUT Q$: IF  LEFT$ (Q$,
     1) = "Y" THEN  GOSUB 1402: GOSUB
     2: GOTO 170
1390  GOSUB 2: GOTO 170
```

To change the number of files the variable NU (lines 0,1) must be changed. Also the boundary ASCII values found in line 95 must be consistent with the number of files. For example, right now every record whose first character has an ASCII value less than or equal to 65 (B%(1)=65) will be placed in MAGFILE1; every record with an ASCII value of 66 to 68 (B%(2)=68) will be put in MAGFILE2 and so on.

If expansion to two disk drives is desired then line 5 should be changed from:

```
5 POKE 43624,1:RETURN
```

to

```
5 POKE 43624,1:IF I > 4 THEN
  POKE 43624,2:RETURN
```

Then any file number greater than 4 (MAGFILE5.to..MAGFILE8) will access drive #2.

The sort routine arranges the records in order, starting at the beginning of the string. This is the easiest way, arranging the data so that the most important information is at the front of the string and the less important near the end. If it is necessary that a sort be done on the magazine name or mo/yr and it is not convenient for that information to be up front (primary sort key), then each individual string within A$ should be a fixed length (padding the end with spaces if it is too short). This will, in effect, create a random access file within each record. Then the information desired (to be sorted) will always be located at the same place in each record. For the present string lengths (40,10,10), to sort on the magazine name, A$ in the sort routine must be replaced with LEFT$ (A$,41,10). For a data sort, A$ must become LEFT$(A$,51,10).

Memory Considerations

Magfiler takes up about 5.4K of space. With eight open files (NU = 8 in lines 0,1), MAXFILES will equal 8 and DOS will set aside 595 bytes for each. This means 4.7K (595*8) will be reserved for disk buffers. In a 48K Apple this leaves 38.4–(5.4+4.7)–2.05(lomem) or about 26K. 300 records* 65 char/record = 19.5K. However, most records will be less than 65 characters and this will leave even more space for program variables and overhead. Maxfiles 16 is the maximum, so NU must be >= to 16. An alternative to increasing maxfiles is close each file after using it. This would increase the number of records that could be handled at once (by about 70), however access time would also increase dramatically. But virtually any number of files could be handled, if you are willing to wait.


```
1401  REM
*** DELETE RENUMBER SUB ********

1402  IF M%(I) = N2 THEN 1420
1403  FOR J = N2 + 1 TO M%(I): GOSUB
      5: PRINT D$;"OPEN";NA$;I;",L
      65"
1405  PRINT D$;"READ";NA$;I;",R";
      J
1407  INPUT A$(J)
1409  NEXT J
1410  PRINT D$
1412  FOR J = N2 TO M%(I) - 1
1414  PRINT D$;"WRITE";NA$;I;",R"
      ;J
1416  PRINT A$(J + 1)
1418  NEXT J
1420  PRINT D$;"WRITE";NA$;I;",RO
      "
1422  PRINT M%(I) - 1
1424 M%(I) = M%(I) - 1:M = M - 1
1426  PRINT D$;"CLOSE": RETURN
1427  REM
**********************************
1429  REM
**********************************
1430  REM
*** SORT MODULE  ***************

1440  PRINT : FLASH : PRINT "SORT
      ING": NORMAL
1441  GOSUB 4
1442  FOR I = 1 TO NU: IF M%(I) <
      = 1 THEN  NEXT I: GOSUB 2: GOTO
      170
1445  FOR J = 1 TO M%(I): GOSUB 5

1446  PRINT D$;"OPEN";NA$;I;;",L6
      5"
1448  PRINT D$;"READ";NA$;I;",R";
      J
1449  INPUT A$(J)
1450  NEXT J
1452  GOSUB 1500
1453  FOR J = 1 TO M%(I): GOSUB 5
1454  PRINT D$;"OPEN";NA$;I;",L65
      "
1456  PRINT D$;"WRITE";NA$;I;",R"
      ;J
1458  PRINT A$(J)
1460  NEXT J: NEXT I: GOSUB 2: GOTO
      170
1470  REM
*** ACTUAL SORT SUB *************

1500 G = M%(I): PRINT NA$;I
1520  IF G <  = 1 THEN  RETURN
1530 G =  INT (G / 2):M4 = M%(I) -
      G
1550 F = 0: FOR K = 1 TO M4:P = K
      + G: IF A$(K) <  = A$(P) THEN
      1593
1585 T$ = A$(K):A$(K) = A$(P):A$(
      P) = T$:F = 1
1593  NEXT K: IF F > 0 THEN 1550
1595  GOTO 1520
1596  REM
**********************************
1598  REM
*** # OF RECORDS MODULE ********

1600 M = 0: HOME : PRINT : FOR I =
      1 TO NU: PRINT NA$;I;"  ";M%
      (I);: HTAB 15: PRINT "RECORD
      S";: HTAB 26: PRINT "ASC";: HTAB
      30: PRINT B%(I - 1) + 1;"-";
      : HTAE 35: PRINT B%(I):M = M
      + M%(I): NEXT : PRINT
1601  : PRINT "TOTAL =  ";M;: HTAB
      15: PRINT "RECORDS"
1602  HTAB 1: VTAB 20: PRINT "HIT
      'RETURN' FOR MENU": GET G$
1605  IF  ASC (G$) = 13 THEN 170
1607  GOTO 1602
1610  REM
**********************************
1620  PRINT : GOSUB 4: FOR I = 1 TO
      NU: PRINT D$;"CLOSE";NA$;I: NEXT
      I
```

MAGFILER

A situation could arise which may result in using up memory inefficiently which is illustrated in TABLE II.

TABLE II

MAGFILE1	20 RECORDS
MAGFILE2	130 RECORDS
MAGFILE3	300 RECORDS
MAGFILE4	140 RECORDS
MAGFILE5	160 RECORDS
MAGFILE6	170 RECORDS
MAGFILE7	147 RECORDS
MAGFILE8	173 RECORDS

This shows a bad choice for the boundaries and now more records cannot be entered in MAGFILE3 while the other files are not full. To solve this difficulty of uneven loading of files, a short program is given in listing 2, called REASSIGN which will re-assign the boundaries and transfer data from the current files to a new set of files. The number of files can even be changed.

Reassign

Reassign will read each text file on the original diskette. Then, based on inputted new boundary values, it will convert them and transfer them to another diskette. It will work on either a single or dual drive system.

The program is self-prompting and will ask for the number of old files, how many new files are planned, what is the base name of the files, and how many drives are available. Then it is necessary to enter the ASCII boundary values for the new files, when asked. REASSIGN will then make the transfer using the new boundary values as the criteria for where each record goes. When the transfer is completed, the new files will have all the information contained in the old files. To access the new files, a copy of MAGFILER should also be put on the new diskette and the variable NU (in lines 0,1) and the boundary values (B% in line 95) must be changed to agree with the new files. *If this is not done, MAGFILER will not be able to work correctly with the new files.*

How Reassign Works

Many of the same subroutines that are used in MAGFILER are also present in REASSIGN. The DOS ON/OFF subroutines are found in lines 20,40 of listing 2. The subroutine at 490–530 determines the indexes and is identical to lines 474–479 in MAGFILER.

Lines 190–270 set up the new files and initialize them with a zero in record 0 of each one.

Lines 290–380 read all the old files record 0's to see how many records are in each. The results are stored in M%(I).

Lines 390–440 reads the data in the first files and stores it in the array A$( ).

Lines 460–540 sets up the indexes to point to the new files (A%( )).

Lines 550–560 actually writes the data to the new files. This process is then repeated for the next old file.

Finally, lines 690–750 write (to record 0 of each new file) a number which indicates how many records are in each file.

If the data base takes up more than one diskette, then the process must be repeated for each diskette.

Table III presents a list of the most important variables for use in modifying or enhancing the programs. Many of the variables used in MAGFILER are the same in REASSIGN.

TABLE III

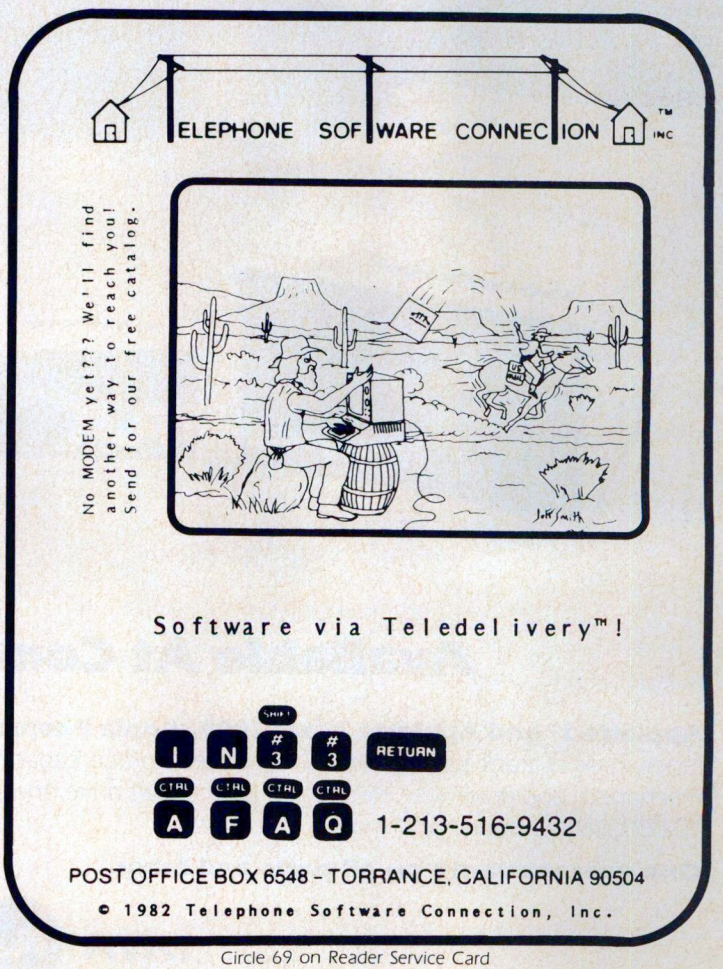
Variable	Function
NU	# of Files in Data Base
NA$	Base Name of Data Base
D4,D5,D6,D7	DOS On/Off Sub Constants
SE	# of Entries Before Automatic Storage to Diskette
B%(I)	ASCII Boundary Value
M%(I)	# of Records in File I
M	Total # of Records in Data Base
V5	Seletion Made From Menu
A$(I)	Record I of Current File
A%(I)	File No. Where Rec. I Belongs

```
]LIST

5  REM

*********************************
* REASSIGN * BY  JACK OLIVIERI *

* CALL -A.P.P.L.E. * MAY, 1983 *
*********************************

10 Z4 =  PEEK (54):Z5 =  PEEK (55
    ):Z6 =  PEEK (56):Z7 =  PEEK
    (57): GOTO 50
20  IF Z4 = 240 THEN  RETURN
30  POKE 54,240: POKE 55,253: POKE
    56,27: POKE 57,253: RETURN
40  POKE 54,Z4: POKE 55,Z5: POKE
    56,Z6: POKE 57,Z7: RETURN
50  INPUT "HOW MANY FILES ARE THE
    RE PRESENTLY ? ";OL
60  INPUT "HOW MANY NEW FILES WIL
    L THERE BE ? ";NU
70  IF OL >  = NU THEN XF = OL
80  IF NU >  = OL THEN XF = NU
90  PRINT  CHR$ (4);"MAXFILES ";X
    F: GOTO 120
100  GOSUB 20: PRINT "INSERT FILE
    S DESTINATION DISK AND HIT
     RETURN": INPUT Z$: GOSUB 40
    : RETURN
110  GOSUB 20: PRINT "INSERT ORIG
    INAL DISKETTE AND HIT RETURN
    ": INPUT Z$: GOSUB 40: RETURN
```

```
120  SZ = 302
130   DIM A$(SZ),A%(SZ),C%(NU + 1)
    ,M%(OL + 1),B%(NU + 1)
140   INPUT "WHAT IS THE 'BASE' NA
    ME OF THE FILES ?   ";NA$
150   INPUT "HOW MANY DISK DRIVES
    ARE AVAILABLE ";DR
160   PRINT : PRINT : FOR I = 1 TO
    NU: PRINT "BOUNDARY #";I;" "
    ;: INPUT B%(I): NEXT I
170   FOR I = 1 TO NU:C%(I) = 0: NEXT

180  D$ =  CHR$ (13) +  CHR$ (4)
190   REM
***  CREATE NU NUMBER OF FILES ON
     DESTINATION DISK ***********
```

```
200   REM
210   IF DR = 1 THEN  GOSUB 100: GOTO
      230
220   POKE 43624,2
230   FOR I = 1 TO NU
240   PRINT D$;"OPEN";NA$;I;",L65"

250   PRINT D$;"WRITE";NA$;I;",R0"

260   PRINT 0: NEXT I
270   PRINT D$;"CLOSE"
280   REM
**********************************


290   REM
*** READ RECORD 0 TO FIND # OF
    RECORDS IN EACH OLD FILE ****


300   REM
310   FOR I = 1 TO OL
320   IF DR = 1 THEN  GOSUB 110
330   POKE 43624,1: GOSUB 20: PRINT
      NA$;I: GOSUB 40
340   PRINT D$;"OPEN";NA$;I;",L65"

350   PRINT D$;"READ";NA$;I;",R0"
360   INPUT M%(I)
370   IF M%(I) = 0 THEN  NEXT I: GOTO
      660
380   REM
**********************************


390   REM
*** READ DATA FROM FILES,
    ONE FILE AT A TIME **********


400   REM
410   FOR J = 1 TO M%(I)
420   PRINT D$;"READ";NA$;I;",R";J

430   INPUT A$(J): NEXT J
440   PRINT D$;"CLOSE"
450   REM
**********************************


460   REM
*** SET UP INDEXES ***************


470   FOR J = 1 TO M%(I): GOSUB 49
      0: NEXT J: GOTO 550
480   REM
*** SUB AT 490 TO SET UP
    INDEXES TO POINT RECORDS
    TO CORRECT FILES *************


490 A%(J) =  ASC (A$(J))
500   FOR K = 1 TO NU
510   IF A%(J) <  = B%(K) THEN A%(
      J) = K: RETURN
520   IF (A%(J) > B%(K)) AND (A%(J
      ) <  = B%(K + 1)) THEN A%(J)
       = K + 1: RETURN
530   NEXT K
540   REM
**********************************


550   IF DR = 2 THEN  POKE 43624,2
      : GOTO 580
560   POKE 43624,1: GOSUB 100
570   REM
*** WRITE DATA TO NEW FILES USING

    INDEXES TO POINT TO FILES ***
```

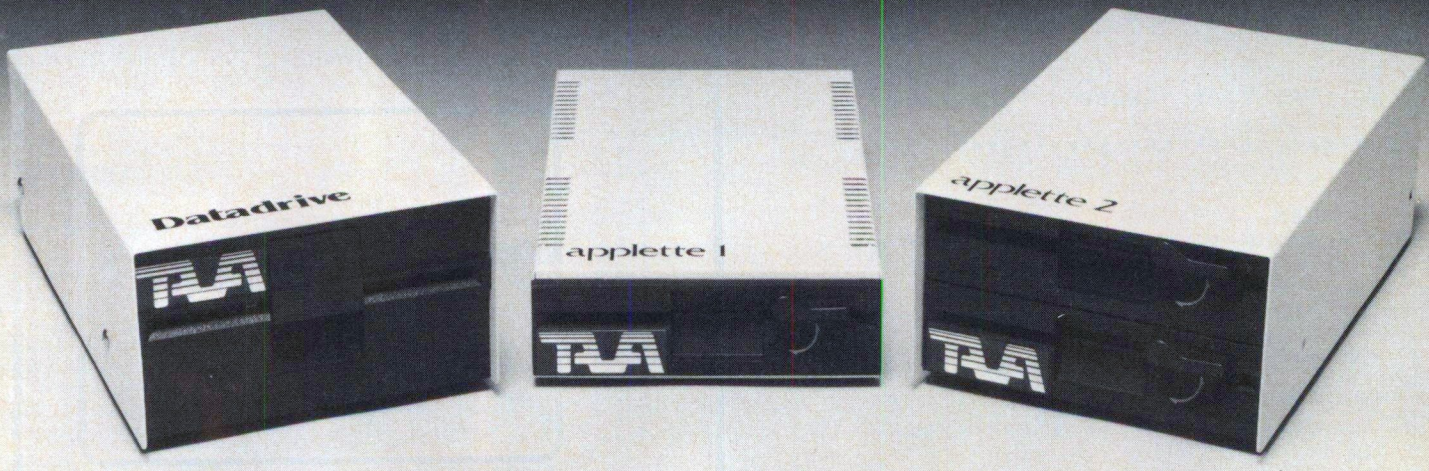
```
580   FOR J = 1 TO M%(I)
590   PRINT D$;"OPEN";NA$;A%(J);;"
      ,L65"
600   PRINT D$;"WRITE";NA$;A%(J);;
      ",R";C%(A%(J)) + 1
610   PRINT A$(J): C%(A%(J)) = C%(A
      %(J)) + 1
620   NEXT J
630   PRINT D$;"CLOSE"
640   REM
**********************************


650   NEXT I
660   IF DR = 1 THEN  GOSUB 100: POKE
      43624,1: GOTO 690
670   POKE 43624,2
680   REM
*** WRITE RECORD 0 TO FILE TO
    SHOW HOW MANY RECORDS IN FILE


690   FOR I = 1 TO NU
700   PRINT D$;"OPEN";NA$;I;",L65"

710   PRINT D$;"WRITE";NA$;I;;",R0
      "
720   PRINT C%(I)
730   NEXT I
740   PRINT D$;"CLOSE"
750   REM
**********************************
```

Circle 69 on Reader Service Card

FROM THE CREATORS OF THE DATA DRIVE® COMES APPLETTE 1® AND APPLETTE 2®

SLIMLINE DRIVES



Available At CompuShack Stores

Applette 1® and Applette 2® are 100% Apple II compatible. Halftracking, DOS, PASCAL, and CP/M®. 300% faster track to track speed with 15% greater storage capacity on a 40 track mode with enhancer diskette. TEAC® mechanism and read/write electronics. Direct shaft drive, metal band positioner, photo coupler write-protected sensor. 10,000 lifetime hours, and more.

One year warranty on all parts and labor.

TAVA CORP.®

HEADQUARTERS/TELEX: 181667—ANSWER BACK: COMPDSHACK IRIN

Circle 37 on Reader Service Card

(714) 261-0200

*DATA DRIVE, APPLETTE 1, APPLETTE 2, and TRUMP CARD are registered trademarks of TAVA Corporation, respectively.

* CP/M is a registered trademark of Digital Research, Inc.
*TEAC is a registered trademark of TEAC Corp.

YOUNGBLOOD:

A Complete Password System

Jason K. Mintz

MY objective in developing this password system was to hide from the user, and DOS, all of the files contained on a diskette until the correct password was entered, at which time the full catalog would be restored.

At the time I began developing the system, I had just received my January 1983 issue of *Call -A.P.P.L.E.*, containing a program written by Ralph Swerdlow, entitled "Password Protection" (p. 13). This program was not adequate for my needs, as it was incredibly easy to bypass the program simply by LOADing it and LISTing it to discover the password, which was contained within the program. I wanted to design a system where: (1) The user has no possibility of gaining access to files on the diskette without knowledge of the correct password, because these files do not appear on the catalog until this password is entered, (2) The user has no choice but to run the password program if he/she has any hopes of gaining this access, and (3) The password is not located in any program that can be listed. However, I must admit I was delighted by the discovery of four POKEs used in "Password Protection", and have used them myself with this system. These POKEs are: POKE 214,128 which prevents the use of [Ctrl-C] to stop a program, and the following POKEs:

POKE 1010,102 : POKE 1011,213 : POKE 1012,112

which run the program in memory whenever the RESET key is pressed, or any BASIC statement is entered. (If you wish to experiment with these, when you are through and want to return your computer to normal, here are their normal values: POKE 214,0 - POKE 1010,191 - POKE 1011,157 - POKE 1012,56)

About the author...

Like father, like son, etc., etc... Jason Mintz is a 15 year old high school student from San Francisco who has had an Apple II Plus for over three years. His father, Ronald Mintz authored the cryptogram program published in the July/August 1981 Call -A.P.P.L.E. *It is not clear to us whether Jason "buys" time on his dad's machine or vice-versa, or whether the household has two Apples. What is clear is that Jason is typical of many teen-agers of the '80s. His program is comprehensive and complete; his understanding (and subsequent explanation) of computer logic outstanding, and the conciseness of his English is on a par with many published authors.*

Welcome to Call -A.P.P.L.E., *Jason!*

While it is possible to install the password system through use of a utility program such as "Disk Zap" or "Diskfixer" which allows the user to read, change or write a track and sector of a disk, the system has been designed for users who do not own such a program. Procedures will also be discussed concerning installation using a utility program for experienced DOS users. Using this type of utility is fine, if you are 100% sure of what you are doing. If you are not, I suggest that you stay away from this if at all possible, in order to reduce the chance of installation error.

To understand how this password system works, you must first understand a bit about Apple's DOS. Your diskette can be compared to a record, with each groove that makes a complete circle once around the record, corresponding to one of the 35 tracks on your diskette (except that tracks do not connect). While track 0 is located at the outermost edges of the disk, the track number increases as you near the center hole and finally reach track 34. Each track is divided into sixteen parts, or sectors, numbered 0 through 15, which contain 256 bytes of information each. The RWTS (READ or WRITE a TRACK and SECTOR) subroutine at $3D9 (consult the DOS manual for further explanation) is what DOS uses when loading or saving a file. During a SAVE, a file is transferred 256 bytes at a time into a file buffer. When this buffer has been filled, its contents are written to a sector of the disk using this subroutine, at which time the next 256 bytes are transferred to a buffer, and the process repeats itself until the entire program has been saved. Programs are loaded in much the same way. A sector at a time is loaded from the disk into a file buffer, until all sectors containing part of the file have been loaded. This password system uses RWTS often, as there is a lot of reading and writing of sectors volved.

The directory of all files contained on a diskette is normally found on track $11, which is set aside for this purpose during initialization. Sector $0 of this track is called the Volume Table of Contents (VTOC). Besides storing here the disk volume number and the track bit maps (these tell DOS which sectors of each track are in use, and which are free), it also contains the track and sector numbers of the first directory sector. Bytes 1 and 2 of the VTOC store these numbers, which ordinarily are set at track $11 and sector $0F respectively. It would seem that an ideal way to create the illusion of an empty catalog would be to fool the computer by resetting these bytes and sidetracking DOS to an alternate catalog when it searches for the diskette's directory. The password

—

A COMPLETE PASSWORD SYSTEM

system uses this idea, changing these bytes from those which correspond to the normal DOS directory at track $11, sector $0F to those which correspond to an alternate catalog created somewhere else on the disk. DOS now sees the alternate catalog as the only catalog on the disk, and it is only when the user inputs the correct password that the system returns the VTOC to normal, thus restoring the catalog to that of the original directory at track $11, sector $0F.

The system is made up of eleven short programs. Nine of them are written in Applesoft, two are binary programs. Although installing the system for the first time may seem like an incredible amount of work, do not be disillusioned for this is not the case. I have tried it myself, and it should take you at most 90 minutes — most likely less.

The alternate catalog, when the system is complete, should contain only these five programs:

*A 003 ENTER	(should be renamed HELLO so that it is run when the disk is booted)
*B 002 HELLO.A	
*B 002 HELLO.B	
*S 004 ENTER.P	
*S 003 EXIT.P	

The regular catalog will contain these four programs:

*A 002 HELLO
*B 002 HELLO.A
*B 002 HELLO.B
*A 004 EXIT

...and these five if the user wishes (they do not need to be on every password diskette, but should be saved on at least one), plus any programs the user wants to store on his/her disk:

*A 004 CHANGE PASSWORD	(use to change the password)
*A 004 INSTALL.1	(these programs install "Password System"
*A 004 INSTALL.2	on to any 3.3 initialized diskette)

```
]LIST

5  REM   *ENTER.P*
10  PRINT "PASSWORD: ";
20  GET A$: IF A$ =  CHR$ (13) THEN
     40
30 P$ = P$ + A$: GOTO 20
40 J =  PEEK (3049) +  PEEK (3050
    ) * 256
50  POKE J,0: POKE 3094,1
60  CALL 3072
65  FOR I = 1 TO 20: IF  PEEK (82
    00 + I) = 0 THEN 70
68  NEXT I
70 V =  PEEK (3090) +  PEEK (3091
    ) * 256
80 C = 1
90  FOR K = V + 8 TO V + 8 + I: IF
     CHR$ ( PEEK (K) - 128) =  MID$
    (P$,C,1) THEN C = C + 1: NEXT
    K: GOTO 110
100  CALL 3561
110  POKE V + 2,15: POKE 3094,2
120  CALL 3072
130 W =  PEEK (3054) +  PEEK (305
    5) * 256
140 X =  PEEK (3059) +  PEEK (306
    0) * 256
150 Y =  PEEK (3064) +  PEEK (306
    5) * 256
160 Z =  PEEK (3069) +  PEEK (307
    0) * 256
170  POKE W,191: POKE X,157: POKE
    Y,56: POKE Z,0
180  HOME
190  PRINT "HELLO !"; CHR$ (7)
200  NEW
```

```
]LIST

5  REM   *EXIT*
10  HOME
20  PRINT
30  PRINT  CHR$ (4);"BLOAD HELLO.
    A"
40  CALL 3051
50  PRINT "PASSWORD: ";
60  GET A$: IF A$ =  CHR$ (13) THEN
     80
70 P$ = P$ + A$: GOTO 60
80 J =  PEEK (3049) +  PEEK (3050
    ) * 256
90  FOR I = J TO J + 9: READ N: POKE
    I,N: NEXT I
100  DATA  0,32,12,0,32,0,0,1,0,0

110  PRINT
120  PRINT  CHR$ (4);"BLOAD HELLO
    .B"
130  CALL 3072
135  FOR I = 1 TO 20: IF  PEEK (8
    200 + I) = 0 THEN 140
138  NEXT I
140 V =  PEEK (3090) +  PEEK (309
    1) * 256
150 C = 1
160  FOR K = V + 8 TO V + 7 + I: IF
     CHR$ ( PEEK (K) - 128) =  MID$
    (P$,C,1) THEN C = C + 1: NEXT
    K: GOTO 180
170  CALL 3561
180  POKE V + 2,2: POKE 3094,2
190  CALL 3072
200 W =  PEEK (3054) +  PEEK (305
    5) * 256
210 X =  PEEK (3059) +  PEEK (306
    0) * 256
220 Y =  PEEK (3064) +  PEEK (306
    5) * 256
230 Z =  PEEK (3069) +  PEEK (307
    0) * 256
240  POKE W,191: POKE X,157: POKE
    Y,56: POKE Z,0
250  PRINT  CHR$ (4);"RUN EXIT.P"
```

***A 003 COPY.1** **(these programs copy "Password System"**
***A 004 COPY.2** **on to any 3.3 initialized diskette)**

How the password system works is this:

When the disk is booted, the alternate catalog is in effect. While the normal catalog is ordinarily located between sectors $0F and $01 of track $11 (the first directory sector is $0F, the second is $0E, third is $0D, etc.), the password system cuts short the normal catalog, ending it at sector $03. The only effect that this has on DOS is that while normally a diskette directory will hold up to 105 files, it now will hold only 91. For most people this should not be a problem. This leaves sectors $01 and $02 available for the alternate catalog, which may store the remaining 14 files.

The purpose of the ENTER program on the alternate catalog (which, as stated previously, should be renamed HELLO) is to change the file types of the programs ENTER.P and EXIT.P from "S" to "A," so that they both may be run successfully without a "FILE TYPE MISMATCH" error. Their file types are set at "S" by the system in order to make them unlistable to unauthorized users. The program loads track $11, sector $02 (this is the first directory sector of the alternate catalog, and contains the directory entries for the programs ENTER.P and EXIT.P) into a buffer at $2000. The user is asked to input a number, which is POKEd into the two addresses in the buffer that control the file types of the two programs. ENTER then rewrites this sector back to disk. If the number entered was 2 or 130 (DOS codes for an Applesoft file type), the file types of ENTER.P and EXIT.P now show on the catalog as "A" and may be run successfully by the password system when they are needed. But if the number entered was not one of the above two, an error message is given and the user may not proceed with the system.

Let us assume, however, that a correct number is entered by the user. The program ENTER.P is automatically run, and the user is asked for the password. ENTER.P then loads the VTOC (track $11, sector $0) into a buffer at $2000 (The system stores the password in the VTOC — bytes 7 through 38 are not used by DOS — beginning with byte 8). The password the user gives is compared with the one in the VTOC. If an incorrect password has been entered, the user is kicked into the monitor. If the correct password has been given, bytes 1 and 2 of the VTOC are changed from $11, $02 (track and sector location of the alternate catalog) to $11, $0F (track and sector location of the regular catalog). This restores the regular catalog, and the VTOC is written back to disk.

The user is now free to do what he/she wishes with his/her restored catalog. When the user is ready to leave the computer, all he/she must do is "RUN EXIT." The password is asked for and again the VTOC is loaded. If the password is correct, bytes 1 and 2 of the VTOC are changed from $11, $0F to $11, $02 and the VTOC is written back to disk. This tells DOS that the only catalog on the disk is the alternate catalog. The password system then runs EXIT.P, which is stored on the alternate catalog. This program does the reverse of ENTER, changing the file types of ENTER.P and EXIT.P (itself) back from "A" to "S" so that LOAD, RUN, BLOAD, BRUN and EXEC will all be ineffective. Of course, it is not necessary to know how the password system works — only that it does work. With the system correctly installed, all the user needs to know is that before turning off the computer, he/she must "RUN EXIT." How could a password system be simpler?

```
*BE9.C0E  Program: HELLO.A

0BE9- 0F 0C A2 66 8E F2 03
0BF0- A2 D5 8E F3 03 A2 70 8E
0BF8- F4 03 A2 80 8E D6 00 60
0C00- A9 0C A0 0A 20 D9 03 60
0C08- 00 00 01 60 01 FE 11
```

Table I

```
0BE9-   0F          ???           Tells "Password" where to POKE
0BEA-   0C          ???            missing section of IOB ($0C0F)
0BEB-   A2 66       LDX   #$66
0BED-   8E F2 03    STX   $03F2   POKE 1010,102
0BF0-   A2 D5       LDX   #$D5
0BF2-   8E F3 03    STX   $03F3   POKE 1011,213
0BF5-   A2 70       LDX   #$70
0BF7-   8E F4 03    STX   $03F4   POKE 1012,112
0BFA-   A2 80       LDX   #$80
0BFC-   8E D6 00    STX   $00D6   POKE 214,128
0BFF-   60          RTS
0C00-   A9 0C       LDA   #$0C
0C02-   A0 0A       LDY   #$0A
0C04-   20 D9 03    JSR   $03D9   RWTS Controlling subroutine
0C07-   60          RTS
0C08-   00          BRK
0C09-   00          BRK
0C0A-   01 60       ORA   ($60,X) Beginning of IOB (HELLO.B begins
0C0C-   01 FE       ORA   ($FE,X)  by ending the IOB. The middle
0C0E-   11 00       ORA   ($00),Y  missing section is POKEd
                                   inbetween the two programs.)
```

A COMPLETE PASSWORD SYSTEM

To install "Password System" for the first time:

There are two ways to install the password system, which may be copied to any disk, regardless of whether it is newly initialized or already contains many programs. Save HELLO.A, HELLO.B, ENTER, ENTER.P, EXIT, EXIT.P and CHANGE PASSWORD to any disk (BSAVE HELLO.A, A$BE9, L$27 — BSAVE HELLO.B, A$C19, L$0C) and until installation is completed, refer to this disk as disk A.

You may now continue in one of the following two ways (they both will provide the same results):

1) Save INSTALL.1, INSTALL.2, COPY.1 and COPY.2 to any disk other than disk A (call this disk B). Take the disk you wish to install the system on to, and for now call this disk C. "RUN INSTALL.1," using disk A as your source disk and disk C as your target disk. Then, when installation has completed, "RUN CHANGE PASSWORD".

   The installation and copy programs may be copied from disk B to disk C (the "password" disk) *after* installation, if you wish.

   Note: You may type "RUN CHANGE PASSWORD" only if the program is stored on the disk whose password you want to change. If it is not, "LOAD CHANGE PASSWORD" from the disk it is saved on, and after inserting the disk whose password you want to change, type RUN.

OR:

2) Using a DOS read-and-write utility:
   1. Change the following bytes on your target disk:

      Track $11 Sector $03

      byte 1: from $11 to $00
      byte 2: from $02 to $00

      Track $11 Sector $0

      byte 2: from $0F to $02

   2. Using FID or another copy program, copy ENTER, HELLO.A, HELLO.B, ENTER.P and EXIT.P from your source disk (disk A) to the target disk.

      Rename ENTER with the name HELLO.
   3. Using your read-and-write utility, change the following on the target disk:

      Track $11 Sector $0

      byte 2: from $02 to $0F

   4. Copy HELLO.A, HELLO.B, EXIT and CHANGE PASSWORD from disk A (source disk) to the target disk.
   5. "RUN CHANGE PASSWORD" and enter the password you want to use with the system (you may change it again at any time, simply by RUNning this program. You also may wish to change the password through use of your read-and-write utility. It is stored, as explained previously, in the VTOC beginning at byte 8. However, I do suggest the use of CHANGE PASSWORD as a more convenient option.

To copy "Password System" from a diskette which already contains the complete system, to any initialized diskette:

1) If using "Password" copy programs:
   1. Enter into the regular catalog of your source disk by booting the source disk and entering the password.
   2. "RUN COPY.1".
   3. When copying has completed, "RUN CHANGE PASSWORD" to assign a password to the target disk.

OR:

2) If using a read-and-write utility, do the following:
   1. Enter into the regular catalog of your source disk by booting the source disk and entering the password.
   2. Copy HELLO.A, HELLO.B, EXIT and CHANGE PASSWORD (and the installation and copy programs if you wish) from the source disk on to the target disk, using FID or a similar copy program.
   3. Change the following bytes on the target disk:

      Track $11 Sector $03

      byte 1: from $11 to $00
      byte 2: from $02 to $00

      Track $11 Sector $0

      byte 2: from $0F to $02

   4. Change the following bytes on the source disk:

      Track $11 Sector $0

      byte 2: from $0F to $02

```
*
*C19.C23  Program: HELLO.B

0C19- 60 01 00 00 00 00 00
0C20- 00 01 EF D8
```

Table II

```
0C19-   60          RTS
0C1A-   01 00       ORA   ($00,X) End of IOB
0C1C-   00          BRK
0C1D-   00          BRK
0C1E-   00          BRK           Unused
0C1F-   00          BRK
0C20-   00          BRK   ($EF,X) Device Characteristics
0C21-   01 EF       ORA           Table
0C23-   D8          CLD
```

5. Copy HELLO (which is really the program ENTER renamed with the name HELLO), HELLO.A, HELLO.B, ENTER.P and EXIT.P from the source disk to the target disk.
6. "LOAD CHANGE PASSWORD" from any disk. Insert the target disk into the drive and type "RUN".

Some Final Notes:

1) I have found that when initializing a 3.3 diskette after using "Password System," the new diskette will sometimes hold only seven programs. The way to fix this and restore your diskette to normal, is to change byte 1 and 2 of track $11, sector $0F from $00, $00 (which is what these bytes are sometimes accidentally set to during initialization after "Password" use — I don't know why) to $11, $0E. When these two bytes are set at $00, this tells DOS that the directory has ended, when in fact it continues at track $11, sector $0E. (If anyone can solve the mystery of why this error sometimes occurs, I would be very interested to know — is some area of memory used by DOS being written over ??)

The changing of these bytes may be accomplished in two ways. Either use a read-and-write utility, or do the following:

LOAD EXIT

Delete lines 40–70, 140–250

Add line 150:

150 POKE 3094,2 : CALL 3072

Determine the value of a variable XX by dividing the number of files your diskette will hold before a "DISK FULL" message is given, by seven. Subtract this number from the number of sectors per track that your disk contains (3.3 = 16 sectors). This value will be XX. Then enter these lines, substituting the value you have calculated, for each XX in the following listing:

```
100 DATA XX,32,12,0,32,0,0,1,0,0
140 POKE 8193,17 : POKE
    8194,XX-1
```

...and finally, RUN the program you have created. (You may also wish to save it to disk, in case it should be needed at a future date. Just remember to change the value of XX each time you use the program, if such a change is needed.)

2) Anyone interested in learning about the RWTS subroutine:

Load EXIT
Delete lines 40–70, 130–250

RUN this program, and enter the monitor by typing "CALL -151". After reading the RWTS section of your DOS manual (p. 94), you may wish to study the binary programs used with "Password System," that you have just loaded into memory, by listing $BE9 through $C23. My controlling subroutine is found at $C00, my Input/Output control block (IOB) table is located at $C0A, and my Device Characteristics Table begins at $C20, (all of these addresses correspond exactly to those of the example routine in the RWTS section of the DOS manual).

```
5  REM   *COPY.1*
10  HOME : FLASH : PRINT "COPY.1"
    : NORMAL
20  PRINT : PRINT : PRINT
30  FLASH : PRINT "STOP!";: NORMAL
    : PRINT  CHR$ (7)
40  PRINT : PRINT "IF YOU HAVE NO
    T ENTERED INTO THE": PRINT "
    REGULAR CATALOG OF YOUR SOUR
    CE DISK."
50  PRINT "PLEASE DO SO NOW -- TH
    EN "
60  PRINT : INVERSE : PRINT  SPC(
    40): NORMAL
70  PRINT : PRINT "COPY FROM SOUR
    CE DISK TO TARGET DISK:"
80  PRINT : PRINT "HELLO.A": PRINT
    "HELLO.B": PRINT "EXIT": PRINT
    "CHANGE PASSWORD"
90  PRINT : PRINT "THEN 'RUN COPY
    .2'"
```

```
]LIST

5  REM   *EXIT.P*
10 J =  PEEK (3049) +  PEEK (3050
    ) * 256
20  POKE J,2: POKE 3094,1
30  CALL 3072
40 V =  PEEK (3090) +  PEEK (3091
    ) * 256
50  POKE V + 118,136
60  POKE V + 153,136
70  POKE 3094,2
80  CALL 3072
90  HOME : PRINT "GOODBYE !"; CHR$
    (7)
100  NEW
```

```
5  REM   *COPY.2*
10  HOME : FLASH : PRINT "COPY.2"
    : NORMAL
20  PRINT : PRINT "INSERT SOURCE
    DISK. PRESS ANY KEY.": GET A$
30  PRINT
40  PRINT  CHR$ (4);"BLOAD HELLO.A"
50  PRINT  CHR$ (4);"BLOAD HELLO.B"
60 J =  PEEK (3049) +  PEEK (3050
    ) * 256
70  FOR I = J TO J + 9: READ N: POKE
    I,N: NEXT I
80  DATA  0,32,12,0,32,0,0,1,0,0
90  CALL 3072
100  POKE 8194,2: POKE 3094,2
110  CALL 3072
120  PRINT "INSERT TARGET DISK. P
    RESS ANY KEY.": GET A$
130  POKE 3094,1: CALL 3072
140  POKE 8194,2: POKE 3094,2
150  CALL 3072
160  POKE 3087,3
170  POKE 3094,1: CALL 3072
180  POKE 8193,0: POKE 8194,0
190  POKE 3094,2: CALL 3072
200  HOME : PRINT "COPY THE FOLLO
    WING PROGRAMS TO THE": PRINT
    "TARGET DISK ";: INVERSE : PRINT
    "IN THIS ORDER": NORMAL : PRINT

210  PRINT "HELLO(";: INVERSE : PRINT
    "ENTER";: NORMAL : PRINT " S
    AVED UNDER THE NAME ";: INVERSE
    : PRINT "HELLO";: NORMAL : PRINT
    ")": PRINT
220  PRINT "HELLO.A": PRINT "HELL
    O.B": PRINT "ENTER.P": PRINT
    "EXIT.P"
230  PRINT : PRINT "'LOAD CHANGE
    PASSWORD' FROM ANY DISK.": PRINT
    "INSERT TARGET DISK AND TYPE
     'RUN'."
```

SIXTH FINGER™

The automatic repeat key for your Apple II*

$9.95

Also introducing our new adjustable unit with on/off feature

$12.95

- Adapts your Apple; all keys held down will repeat after the perfect delay.
- Great for program editing, word processing & VisiCalc*
- Easy for anyone to install. Plugs onto the Apple's encoder board (all rev. 7 or later and late rev. 6 with encoder board).

(30 day trial)

Regular unit	$9.95
Adjustable unit	12.95
Add shipping	1.00
Overseas shipping	3.00

Cal. res. add 6% tax

To order or request free instruction sheet use order form or call (213) 675-4706

Amount enclosed $ ____ For ____
City ____ St. ____ zip ____
Address ____
Name ____
Sign. ____
Card # ____ Exp. ____
○ Visa ○ MasterCard ○ Check or money order enclosed ○ Free instruction sheet

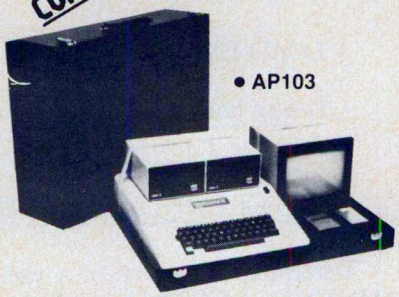
Ivers Specialties
Dept. CA-2
12841 Hawthorne Blvd.
Box 524
Hawthorne, Ca. 90250

Dealer Inquires Invited

*Apple is a registered trademark of Apple Computer Co.
*VisiCalc is a registered trademark of VisiCorp.

Circle 17 on Reader Service Card

Computer Case Company



• AP103

Attache-style cases for carrying and protecting your complete computer set-up. Accommodates equipment in a fully operational configuration. Never a need to remove equipment from case. Simply remove lid, connect power, and operate.

AP101	Apple II with Single Drive	$109
AP102	Apple II with Two Disk Drives	119
AP103	Apple II, 9 Inch Monitor & Two Drives	129
AP104	Apple III, Two Drives & Silentype Printer	139
AP105	13" Monitor with Accessories	99
AP106	AMDEK Color Monitor	119
RS201	TRS-80 Model I, Expansion Unit & Drives	109
RS204	TRS-80 Model III	129
AT301	ATARI Computers with Peripherals	109
P402	Centronics 730/737 & Radio Shack Printer	89
P403	Epson MX70/80 or Microline 82A	89
P404	Epson MX100 Printer	99
P405	IDS 560 or Prism 132 Printer	109
P406	Starwriter/Printmaster F-10 Printer	119
P407	Okidata Microline 83A or 84 Printer	99
P408	Prowriter 2 Printer	99
P409	Prowriter (Apple Dot Matrix) Printer	89
IB501	IBM Personal Computer	129
IB502	IBM Monitor	99
HP601	HP41 with Accessories	99
CM703	Commodore Model 64 with Drives	119
CM704	Commodore Model 64 with Dataset	109
NS010	North Star Advantage	139
CC80	Matching Attache Case (5")	85
CC90	Matching Attache Case (3")	75
CC91	Matching Accessory Case	95
CC92	5.25" Diskette Case	49

Computer Case Company

5650 Indian Mound Court
Columbus, Ohio 43213
(614) 868-9464

CALL TOLL FREE
800-848-7548

Circle 7 on Reader Service Card

PASSWORD

There are only a few parts of my binary programs which differ from those in the RWTS example routine (see tables 1 and 2). Part of my IOB — $C0F through $C18 — is POKEd into memory and is not part of HELLO.A or HELLO.B (see lines 80–100 of the program ENTER). This is done for no other reason except to confuse the unauthorized user, who, out of all the password programs, is only permitted to list ENTER. Therefore, ENTER must be as complicated as possible in order to discourage someone trying to "crack" the system. $BE9 and $BEA tell the system where to POKE the missing section of the IOB. Addresses $BEB through $BFF execute the four POKEs (see paragraph two) which make it impossible for the user to get out of the password programs.

3) Since the only time the HELLO program on the regular catalog is RUN is when the user has forgotten to "RUN EXIT" before leaving the computer, it might be a good idea to have this HELLO program print something like:

> HELLO! YOU FORGOT TO "RUN EXIT".

so that the user is a little more careful the next time.

Now, it's time to get to work. Be sure that before you begin to install the system, you have read and fully understand this entire article. If you have a question about something, go back and read. It is also a good idea to read through the article once again after you have installed the system.

I hope you will be as satisfied with "Password System" as I am. I encourage all Apple owners to experiment with RWTS, and to investigate the inner workings of their DOS. When I first had an idea for developing this system, I knew nothing at all about DOS, but both by reading my DOS manual (especially those sections dealing with RWTS and the diskette directory) and experimenting on my own, I learned a great deal. I had told myself that what I had in mind was impossible. Well, obviously not...

```
5  REM   *CHANGE PASSWORD*
10  HOME
20  PRINT
30  PRINT  CHR$ (4);"BLOAD HELLO.
    A"
40  CALL 3051
50  PRINT "PASSWORD: ";
60  GET A$: IF A$ =  CHR$ (13) THEN
    80
70 P$ = P$ + A$: GOTO 60
80 J =  PEEK (3049) +  PEEK (3050
    ) * 256
90  FOR I = J TO J + 9: READ N: POKE
    I,N: NEXT I
100  DATA  0,32,12,0,32,0,0,1,0,0

110  PRINT
120  PRINT  CHR$ (4);"BLOAD HELLO
    .B"
130  CALL 3072
140 V =  PEEK (3090) +  PEEK (309
    1) * 256
150 C = 1
160  FOR K = V + 8 TO V + 8 +  LEN
    (P$) - 1: POKE K, ASC ( MID$
    (P$,C,1)) + 128:C = C + 1: NEXT
    K
170 L = 25 -  LEN (P$)
180  FOR M = V + 8 +  LEN (P$) TO
    V + 8 +  LEN (P$) + L: POKE
    M,0: NEXT M
190  POKE 3094,2
200  CALL 3072
210 W =  PEEK (3054) +  PEEK (305
    5) * 256
220 X =  PEEK (3059) +  PEEK (306
    0) * 256
230 Y =  PEEK (3064) +  PEEK (306
    5) * 256
240 Z =  PEEK (3069) +  PEEK (307
    0) * 256
250  POKE W,191: POKE X,157: POKE
    Y,56: POKE Z,0
```

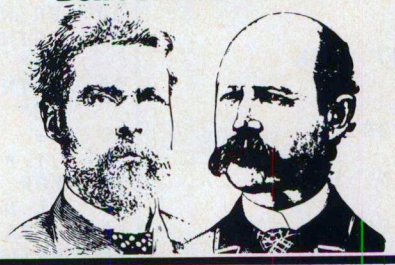
```
5  REM   *ENTER*
10  HOME
20  PRINT
30  PRINT  CHR$ (4);"BLOAD HELLO.
    A"
40  CALL 3051
50  PRINT "NUMBER: ";
60  GET A$: IF A$ =  CHR$ (13) THEN
    80
70 P$ = P$ + A$: GOTO 60
80 J =  PEEK (3049) +  PEEK (3050
    ) * 256
90  FOR I = J TO J + 9: READ N: POKE
    I,N: NEXT I
100  DATA  2,32,12,0,32,0,0,1,0,0

110  PRINT
120  PRINT  CHR$ (4);"BLOAD HELLO
    .B"
130  CALL 3072
140 V =  PEEK (3090) +  PEEK (309
    1) * 256
150  POKE V + 118, VAL (P$)
160  POKE V + 153, VAL (P$)
170  POKE 3094,2
180  CALL 3072
190  PRINT  CHR$ (4);"RUN ENTER.P
    "
```

```
5  REM   *INSTALL.1*
10  HOME : FLASH : PRINT "INSTALL
    .1": NORMAL : PRINT
20  PRINT "INSERT SOURCE DISK. PR
    ESS ANY KEY.": GET A$
30  PRINT
40  PRINT  CHR$ (4);"BLOAD HELLO.
    A"
50  PRINT  CHR$ (4);"BLOAD HELLO.
    B"
60 J =  PEEK (3049) +  PEEK (3050
    ) * 256
70  FOR I = J TO J + 9: READ N: POKE
    I,N: NEXT I
80  DATA  2,32,12,0,32,0,0,1,0,0
90  PRINT "INSERT TARGET DISK. PR
    ESS ANY KEY.": GET A$
95  POKE 3087,3
100  CALL 3072
110  POKE 8193,0: POKE 8194,0
120  POKE 3094,2: CALL 3072
130  POKE 3094,1: POKE 3087,0
140  CALL 3072
150  POKE 8194,2: POKE 3094,2
160  CALL 3072
170  HOME : PRINT "COPY THE FOLLO
    WING PROGRAMS ON TO THE": PRINT
    "TARGET DISK ";: INVERSE : PRINT
    "IN THIS ORDER": NORMAL : PRINT

180  PRINT "ENTER": PRINT "HELLO.
    A": PRINT "HELLO.B": PRINT "
    ENTER.P": PRINT "EXIT.P": PRINT

190  PRINT "RENAME ";: INVERSE : PRINT
    "ENTER";: NORMAL : PRINT " W
    ITH THE NAME ";: INVERSE : PRINT
    "HELLO": NORMAL
200  PRINT : PRINT "THEN 'RUN INS
    TALL.2'
```

```
5  REM   *INSTALL.2*
10  HOME : FLASH : PRINT "INSTALL
    .2": NORMAL : PRINT
20  PRINT "INSERT SOURCE DISK. PR
    ESS ANY KEY.": GET A$
30  PRINT
40  PRINT  CHR$ (4);"BLOAD HELLO.
    A"
50  PRINT  CHR$ (4);"BLOAD HELLO.
    B"
60 J =  PEEK (3049) +  PEEK (3050
    ) * 256
70  FOR I = J TO J + 9: READ N: POKE
    I,N: NEXT I
80  DATA  0,32,12,0,32,0,0,1,0,0

90  PRINT "INSERT TARGET DISK. PR
    ESS ANY KEY.": GET A$
100  CALL 3072
110  POKE 8194,15
120  POKE 3094,2
130  CALL 3072
140  HOME : PRINT "COPY THE FOLLO
    WING PROGRAMS": PRINT "TO TH
    E TARGET DISK:": PRINT
150  PRINT "HELLO.A": PRINT "HELL
    O.B": PRINT "EXIT": PRINT "C
    HANGE PASSWORD"
160  PRINT
170  PRINT "THEN 'RUN CHANGE PASS
    WORD' TO ASSIGN A": PRINT "P
    ASSWORD TO YOUR TARGET DISK.
    "
```

NEW Beagle Bros Unlocked Utilities!

All Beagle disks are Apple II/IIe compatible, and COPYABLE. Don't settle for less!



Beagle Bros™ Micro Software Inc.

4315 SIERRA VISTA / SAN DIEGO, CA 92103
619-296-6400

"APPLE" is a registered trade mark of You-Know-Who.

NEW! DOUBLE-TAKE

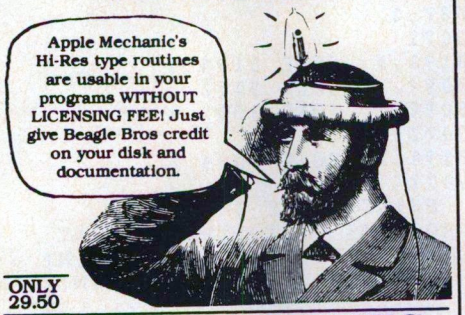
ONLY 34.95

2-WAY-LIST/MULTIPLE UTILITY
BY MARK SIMONSEN

LISTS & CATALOGS SCROLL UP and DOWN, making file names & program lines much easier to access. Arrow-keys control scroll direction. NEW LIST FORMAT lists each program statement on new line— FAST program trace/de-bugging.

VARIABLE-DISPLAY shows all of a program's strings & variables with current values. CROSS-REFERENCE shows line nos. on which each variable/string appears. Better RENUMBER/APPEND lets you merge programs (not just end-to-end). Applesoft AUTO-LINE-NUMBER as you type. Instant Hex/Dec Converter, Program Stats, Cursor Eliminate/Redefine, Free Space-On-Disk...

(includes Peeks/Pokes Chart)



ONLY 29.50

APPLE MECHANIC

SHAPE-WRITER/BYTE-ZAP UTILITY
BY BERT KERSEY

SHAPE EDITOR: Keyboard-draw shapes for hi-res animation in your programs. Design Proportionally-Spaced TYPEFACES with special characters. Six fonts on the disk. Listable demos show how to use shape tables to animate games, graphics and professional Charts & Graphs.

BYTE-ZAP: Rewrite any byte on a disk for repair or alteration. Load entire sectors on the screen for inspection. Hex/Dec/Ascii displays and input. Complete instructions & experiments for making trick file names, restoring deleted files, etc.

MORE: Useful music, text and hi-res tricks for your programs. Educational documentation.

(includes Peeks/Pokes Chart & Tip Book#5)

TYPEFACES

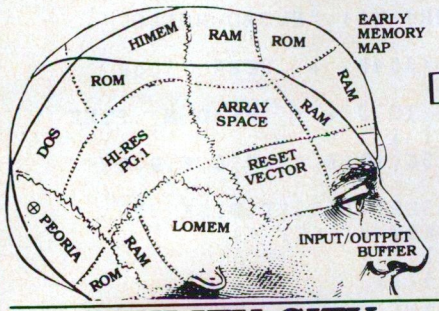
ONLY 20.00

FOR APPLE MECHANIC

26 NEW FONTS for Apple Mechanic programs. All sizes of fully-editable characters.

BEAGLE MENU: Display only filenames you want from YOUR DISKS (e.g. only Applesoft or only Locked files) for one-key selection.

(includes Peeks/Pokes Chart & Beagle Menu Utility)



UTILITY CITY

ONLY 29.50

21 UTILITIES ON ONE DISK
BY BERT KERSEY

LIST FORMATTER prints each program statement on a new line. Loops indented with printer page breaks. A great de-bugger! Also...

MULTI-COLUMN catalogs for printouts, auto-post Run-number & Date in programs, put invisible commands in programs, create INVISIBLE file names, alphabetize/store info on disk, convert decimal to hex or INT to FP, renumber to 65535, append programs, dump text-screen to printer...

MORE TOO: 21 Programs Total, a best-seller!

(includes Peeks/Pokes Chart & Tip Book#3)

DOS BOSS

ONLY 24.00

DISK COMMAND EDITOR
BY BERT KERSEY & JACK CASSIDY

RENAME COMMANDS & ERROR MESSAGES: "Catalog" can be "Cat"; "Syntax Error" can be "Oops" or anything you want. Protect your programs; unauthorized save-attempt can produce "Not Copyable" message. Also LIST-prevention and one-key program-run from catalog.

CUSTOMIZE DOS: Change Disk Volume heading to your message. Omit/alter catalog file codes. Fascinating documentation and tips; hours of juicy reading and Apple experiments.

ANYONE USING YOUR DISKS (booted or not) will be formatting DOS the way you designed it.

(includes Peeks/Pokes Chart & Tip Book#2)

NEW! FLEX TEXT

ONLY 29.50

20/40/56/70-COLUMNS—NO HARDWARE!
BY MARK SIMONSEN

PRINT VARIABLE-WIDTH TEXT on the hi-res screens with normal Applesoft commands (including Htab 1-70). Normal, expanded & compressed text on same screen— no hardware!

ADD GRAPHICS TO TEXT or vice-versa. Run existing programs under Flex Text control. Easy to use and compatible with PLE© and GPLE.©

DOS TOOL KIT© FONT compatibility, or use Flex Text fonts. Select up to 9 fonts with ctrl-key commands. Print/List/Catalog in any style! Custom TEXT CHARACTER EDITOR included.

70-Columns requires Monitor, not TV; (includes Peeks/Pokes)

BEAGLE BAG

ONLY 29.50

12-GAMES-PLUS ON ONE DISK
BY BERT KERSEY

COMPARE BEAGLE BAG with any one-game locked-up game disk on the market today. All 12 games are a blast, the price is right, the instructions are crystal clear, AND the disk is copyable. You can even change the programs or list them to LEARN, and see what makes them tick.

TWELVE GAMES from the Applesoft Ace, Bert Kersey— TextTrain, Slippery Digits, Wowzo, Buzzword, Magic Pack... A GREAT VARIETY of games that tap your Apple's flexibility. Excellent review in January 83 Softalk (page 148).

BEAGLE MENU TOO: See "Typefaces" disk.

(includes Peeks/Pokes Chart & Beagle Menu Utility)

FRAME-UP

ONLY 29.50

HIGH-SPEED DISPLAY UTILITY
BY TOM WEISHAAR

MAKE PROFESSIONAL PRESENTATIONS of existing hi-res, lo-res & text frames. FAST hi-res loads in 2½-seconds! Paddle or Keyboard-advance frames. UNATTENDED SHOWS optional— each pic arranged & pre-programmed to display from 1 to 99 seconds. TEXT SCREEN EDITOR lets you create b/w text "slides". Add type "live" from keyboard during shows. Send copies of presentations-on-disk to your friends and associates.

(includes Peeks/Pokes Chart)

NEW! ProntoDOS

ONLY 29.50

HIGH-SPEED DISK UTILITY
BY TOM WEISHAAR

Put HIGH-SPEED DOS in your Apple's normal memory, Language Card or IIe high-memory—

Function	Normal	Pronto
BLOAD HI-RES IMAGE	10 sec.	3 sec.
LOAD 60-SECTOR PROGRAM	16 sec.	4 sec.
SAVE 60-SECTOR PROGRAM	24 sec.	9 sec.
BLOAD LANGUAGE CARD	13 sec.	4 sec.

(TEXT FILES: No Change)

BOOT PRONTO disk or your updated disks. Create new high-speed disks with normal INIT command. Compatible with ALL DOS COMMANDS and almost ALL of your programs.

LANGUAGE CARD or IIe high-memory can hold DOS— 10,000 Extra-Bytes of program space!

MORE DISK SPACE: ProntoDOS frees-up 15-extra-sectors per disk, almost one full track!

PLUS: Auto-Free-Disk-Space, New "TYPE" Command displays Text Files, and much more...

(includes Peeks/Pokes Chart)

ALPHA PLOT

ONLY 39.50

HI-RES GRAPHICS/TEXT UTILITY
BY BERT KERSEY & JACK CASSIDY

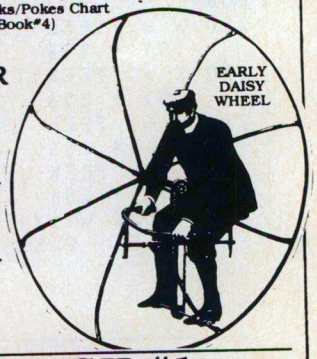
DRAW IN HI-RES, on 2 pages, using keyboard OR paddles/joystick. View lines before plotting. Mixed-colors and reverse (background opposite). Fast circles, boxes and ellipses; filled or outlined.

COMPRESS HI-RES PICS TO 1/3 DISK-SPACE. Superimpose pages or re-locate any rectangular image area anywhere on either hi-res page.

HI-RES TEXT: Proportional spacing, adjustable character size and color, upper & lower case, no htab/vtab limits, sideways typing for graphs.

(includes Peeks/Pokes Chart & Tip Book#4)

GOTO YOUR Apple Software Store for Beagle Bros disks. If they don't have what you want, tell them to GET ON THE STICK by phoning Beagle Bros, 619-296-6400, or ANY Apple SOFTWARE DISTRIBUTOR.



TIP DISK #1

ONLY 20.00

100 TIP BOOK TIPS ON DISK
BY BERT KERSEY

100 LISTABLE PROGRAMS from Beagle Bros Tip Books 1-4. Make your Apple do things it's never done! All programs changeable for experimentation. Includes our Apple Command Chart with ALL Applesoft, Integer & DOS Commands!

(includes Peeks/Pokes AND Apple Command Charts)

RUSH the following disks by First Class Mail:

☐ Alpha Plot	39.50	☐ ProntoDOS	29.50
☐ A.Mechanic	29.50	☐ Tip Disk#1	20.00
☐ Beagle Bag	29.50	☐ Typefaces	20.00
☐ DOS Boss	24.00	☐ Utility City	29.50
☐ Double-Take	34.95		
☐ Flex Text	29.50	☐ Add me to mail list.	
☐ Frame-Up	29.50	☐ I'm already ON it.	

AT YOUR APPLE DEALER NOW!
Or order directly from Beagle Bros—

VISA master charge

Visa/MasterCard/COD, call TOLL-FREE
Nationwide: 1-800-854-2003 ext. 827
California: 1-800-522-1500 ext. 827
Alaska/Hawaii: 1-800-854-2622 ext. 827

OR mail U.S.Check, Money-Order or Visa/MC# to BEAGLE BROS, 17th-Floor 4315 SIERRA VISTA, SAN DIEGO, CA 92103

Add 1.50 First Class Shipping, any-size order. Overseas add 4.00. COD add 3.00. California add 6%. ALL ORDERS SHIPPED IMMEDIATELY

Circle 3 on Reader Service Card

A **Call -A.P.P.L.E.** TECHNOTE:

APPLESOFT CAN THROW YOUR VALUABLES IN THE TRASH CAN

Cornelis Bongers/Call -A.P.P.L.E. staff writer

RECENTLY, one of our students spent several days in finding a bug in a large program. The program made a heavy use of strings and when the problem was brought to my attention, my first guess was that it probably was caused by the student himself, trying to reset his way out of the garbage collection delay. As it turned out, I was wrong.

The program listed in figure 1 creates a situation in which the bug shows up. A little study of the program will reveal that it is a perfectly normal program; yet things don't work as they should. What you might have expected is true indeed. We have caught a new Applesoft bug and it looks as if this one may be one of the nastier bugs found so far.

figure 1

```
5   REM ***** SO WHAT
      DID YOU EXPECT
      *****
10  REM APPLESOFT GARBAGE
      COLLECTION BUG
20  REM HIMEM IS SUPPOSED
      TO BE $9600
100   DIM A(7208)
110   FOR I = 1 TO 15
120   READ A$:B$ = B$
      + " " + A$
130   PRINT B$
140   NEXT
150   DATA C,A,L,L,A,P,P,L,E,
      I,S,G,O,O,D
]RUN
 C
 C A
 C A L
 C A L L
 C A L L A
 C A L L A P
 C A L L A P P
 C A L L A P P L
 C A L L A P P L E
 C A L L A P P L E I
 C A L L A P P L E I S
                     G
                       O
                         O
                           D
```

The bug is in Applesoft's garbage collection routine and it may cause a real mess in your stringpool. Whether or not the bug will occur depends on the (part of the) statement Applesoft is executing at the time the garbage collection routine is invoked. As long as garbage collection is caused by non-string operations, you will have no problems. However, if garbage collection is done because — during the evaluation of a string expression — free memory turns out to be insufficient to continue execution, there is a good chance you are in trouble.

To understand the cause of the bug, two matters need some further explanation. The first involves the way Applesoft processes string expressions. Just like when evaluating arithmetic expressions, Applesoft uses a stack oriented approach for the evaluation of string expressions. However, temporary results are not only stored on the 6502 stack, but also on a separate string descriptor stack on zero-page (locations $55-$5D). This stack consists of three cells of three bytes each. Each cell provides room for one string descriptor; the first byte contains the length of the string and the other two bytes a pointer to the string. Furthermore, one zero-page location ($52) is used as a stack pointer and it points always to the next empty cell on the stack. Another zero-page location ($53) points to the last string descriptor pushed on the stack. The zero-page string descriptor stack is primarily maintained to keep track of active temporary strings when evaluating a string expression. This information is needed for the garbage collection routine, which treats these temporary strings just like ordinary strings. Furthermore, the descriptor stack is used to retrieve information regarding the storage area of strings that are no longer needed, because the memory occupied by these strings can be given free. If the evaluation of a string expression is finished, the string descriptor stack will be empty.

The second matter concerns the garbage collection algorithm. As already outlined in several articles, Applesoft's garbage collection routine is not very efficient. Delays of ten minutes or more may occur when the garbage collection routine is invoked. This is also the reason that separate garbage collection routines have been developed. These routines can be linked to BASIC with a CALL or & statement and they speed up garbage collection considerably.

The algorithm Applesoft uses works as follows. When Applesoft discovers there is not enough room left to dimension an array, create a variable or make a new string, a call is done to the garbage collection routine, which starts at $E484. This routine searches through three separate groups of string descriptors; the string descriptors of the temporary strings that are in process (these are stored on the string descriptor stack on zero-page), the string descriptors associated with (and stored in) the string variables in the variable space and the string descriptors associated with the arrays in the array space. The purpose of the search is to find the string that is closest to HIMEM. After all three groups have been searched, the string closest to HIMEM is moved up, so that its end is just below HIMEM and the corresponding string descriptor is then adjusted accordingly. Note that strings that are overwritten by the move must be inactive strings (garbage) for there are no string descriptors associated with them.

Next, the three groups of string descriptors — excluding the string descriptor of the string that just has been moved — are searched again for the string closest to HIMEM and this string is moved up to just below the first string. This process continues until all active strings reside as a compact block at the end of free memory. As outlined in a recent *MICRO* article, this is essentially a quadratic process, which explains its inherent slowness, especially when the number of active strings is large. A garbage collection routine based on a linear algorithm presented in the article (also available in *Ampersoft*), demonstrates that garbage collection time can be reduced by a factor 80 (if 1000 strings of length 15 are active) to 600 (if 5000 strings of length 3 are active). But let us not get carried away by the interesting aspects of garbage collection and instead concentrate on the bug. We are relatively close now.

VALUABLES IN THE TRASH CAN

When Applesoft is searching through the three groups of descriptors, it needs some end-search indicator. After some time all strings are namely moved up and then the garbage collection is ready. Applesoft applies a simple trick to see whether the garbage collection is finished. Each time the three groups are searched, the result of the search, which consists of the pointer to the string descriptor associated with the "current highest" string, is stored in $8A (low) and $8B (high). Now, before the new search is started, Applesoft stores a zero in $8B. After the search is completed, a check is done to see if $8B is still zero. If not, it means that another string has been found. In that case, this string is moved up and the garbage collection process continues. On the other hand, if $8B contains a zero, all strings are apparently moved up (remember that all strings that already have been moved up are excluded from the search), so Applesoft decides garbage collection is finished and returns control to the caller.

If you read the foregoing carefully, the cause of the bug should be clear. The clue is that location $8B, which serves as end-search indicator, is also used to store the high byte of the pointer to the string descriptor associated with the (current) string that is closest to HIMEM. *But if this string descriptor resides on the string descriptor stack on zero-page, the high byte of this pointer will also be zero.* Hence, although Applesoft thinks garbage collection is finished, this need not be the case, since there may be one or more strings (that were in process) left which should also be moved up.

This explains why the demo program in figure 1 led to such strange results. Since the garbage collection routine is invoked while Applesoft is evaluating a string expression, some temporary strings, whose descriptors are saved on the string descriptor stack, will not be moved up. At a later stage in the same evaluation, these strings are then overwritten by other strings, resulting ultimately in a nonsense assignment to B$.

Fortunately, there are several methods to circumvent the bug. The easiest is to change the end-search indicator Applesoft uses to detect if garbage collection is ready. This solution is — of course — only applicable to those who have Applesoft in RAM or Eprom. It turns out that only two bytes need be changed from $8B to $91. The first change must be made at address $E48F and the second at $E563. After these modifications are made, Applesoft will use location $91 as its end-search indicator.

This location is set to zero upon the start of the search and if a string that needs be moved up is found, $91 is set to a value that differs from zero (by the search routine). At the end of the search Applesoft will check $91. If it contains a zero, garbage collection is terminated, otherwise it continues.

Note: Enter figure 1 exactly as shown, no space after DATA or REM. If GPLE is in 48k memory, DIM A (6389)

The second method to avoid the bug is to force garbage collection regularly by inserting A=FRE(0) statements in the critical parts of your program. If garbage collection is forced in this way, there will be no temporary strings active and thus you don't run the risk that some of these will not be processed properly. If you don't look forward to repeated garbage collection, there are two other things you can do. The first is to use one of the fast garbage collection routines (see among others, Fast Garbage Collection by Randy Wigginton, *Call -A.P.P.L.E.*, Vol IV, Number 1 and Straightforward Garbage Collection by Cornelis Bongers, *MICRO*, August 1982). The alternative is to postpone garbage collection as long as possible and to execute the A=FRE(0) statement only if you no longer can avoid it. How this can be accomplished is outlined in Weed Out the Trash by Clay Ruth, *Call -A.P.P.L.E.*, Vol V, Number 11.

Take a PEEK at these two A.P.P.L.E. specials

A.P.P.L.E. Members Only

★Peeking Special

Get:	Peeking I (1978)	$7.50
	Peeking II (1979)	$10.00
	Peeking III (1980)	$15.00
	regular price	$32.50
	NOW:	$20.00

Overseas shipping wt: 3 lbs.

★Annual Special

Get 1981 and 1982 bound volumes.

regular price	$37.00
special price	$25.00

Overseas shipping wt. 9 lbs

A.P.P.L.E. ORDERS
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245

Washington State Residents add 7.9% sales tax

Offer expires June 30, 1983

AN ECONOMIC AID:

AUTOMATED TYPESETTING

Mike Shadick

AT a time when many businesses are going through an agonizing process of belt-tightenings, a few are taking an opposite approach.

That's all well and good, if you happen to be in a so-called "growth" industry, such as microcomputers, right? But what if one's firm is centered in an industry which is at least as depressed as the economy at large, if not more so? How does one see fit to expand, *then?*

Cold Type Setters, Inc., of Minneapolis, is in just such an industry. Business in the printing and typography trades is generally, to put it mildly, rather "light" these days, especially in the Twin Cities, traditionally one of the nation's leading publishing centers. Yet in the past six months, Cold Type has effectively *tripled* its typesetting capacity, by doubling the shop's floorspace and adding a number of state-of-the-art computerized typesetting units which look like equipment straight out of "Star Wars."

How has Cold Type seen fit to finance this expansion? And, even more pertinently, *why have they even bothered?*

"It was a matter of either sitting back and letting the economic slump hit us hard," explains Cold Type President Wally Simpson, "or hit back at it even harder. If the economy were stronger, we'd probably be expanding — but much more slowly. With things as they *are,* rather than how we'd *like* 'em to be, we figured that we had no real choice in the matter!"

An unorthodox business philosophy? And one destined for a date with disaster?

Not in Cold Type's case. For, by adding to its equipment and staff, Cold Type has expanded its abilities to tap into a relatively new and rapidly-growing area of the printing/typography trades. Namely: *remote* typesetting.

Also known as "setting type by telephone."

Basically, what it means is that the first (and most labor-intensive) step in the typography process — the actual keyboarding of the copy to be typeset — is done by the client, instead of by the typography house. If a client already has word processing equipment, the client's copy can be keyboarded on that equipment. Or, the client can lease one or more specially-adapted WP microcomputers from Cold Type.

"The savings that we realize," Wally Simpson notes, "both in terms of time and labor costs, through clients keyboarding their own copy, are quite substantial. We can then, of course, pass those savings back to the client, in the form of lower cost per galley page of finished type. Perhaps needless to say, our clients are wildly enthusiastic about this new service!"

Remote keyboarding works either "directly" — that is, by Cold Type receiving 5¼-inch floppy disks containing data files of the client-keyboarded copy — or by telephone, using a modulator/demodulator (Modem) data-transmission system. "No special phone lines are required," Simpson notes. "We have remote-keyboarding clients throughout the nation."

Interestingly enough, low cost is only one benefit of remote typesetting. There are definite speed and quality benefits, as well. "Nobody knows a client's products or services better than the client," Simpson observes. "So when they keyboard their own copy — be it for a price list, a brochure, a newsletter, or whatever — it gets keyboarded exactly as *they* want. After all, typography which pleases the client is the name of the game. And when our clients perform the first step themselves, they are helping to, in effect, *guarantee* their own satisfaction with the final product from our shop."

Cold Type's remote-typesetting clients include a number of national publications, and a regional "weekly," published by the Minnesota State Legislature, when in session. Copy for *Wrap-up,* the Legislature's slim but information-packed periodical, is keyboarded by Legislative secretaries on Intertec SUPERBRAIN word-processing equipment leased from Cold Type. Data files of the keyboarded copy are then sent by phone to Cold Type's Washington Ave. shop, where computer-coded typesetting commands are automatically added. These commands control such

—●→

CALL -A.P.P.L.E. readers and members may be interested in knowing that their magazine was if not the first, then certainly among the pioneers in automated typesetting. Going back some 18 months to September, 1981, *Call -A.P.P.L.E.* has been using a D. C. Hayes *Micromodem II* to telecommunicate our story files to an RS-232 terminal connected to a Compugraphics *Editwriter 7500,* owned by *CompuSystems,* a Seattle typesetting firm co-owned by Mike Hughes.

Through the use of a simple "translation table" sent at the beginning of each day's transmission, *Call -A.P.P.L.E.* word processing personnel actually keystroke in all typesetting commands that the Editwriter requires, much the same as the formatting commands you would insert into any popular word processor file, thus when the telecommunicated copy is received by CompuSystems, very minimal typographer intervention is required, substantially reducing (to the member's benefit) our overall production costs.

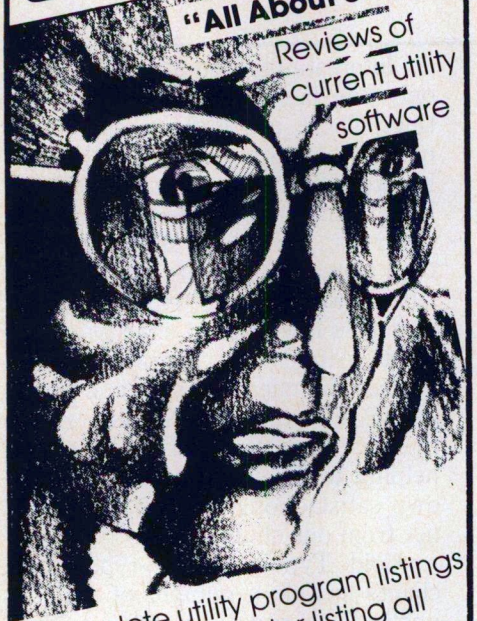
CompuSystems' Mike Hughes, as well as *Call -A.P.P.L.E.* pioneered this field, and as a consequence, Compugraphics, the Boston firm which manufactures the Editwriter 7500, refers many of their Apple and telecommunications troubleshooting calls directly to Mike.

When bites aren't enough,

core gives you more.

"All About Utilities"

Reviews of current utility software



Complete utility program listings also a foldout poster listing all utility data.

CORE, the quarterly that fills its pages with all the information on a single topic. **CORE** carries expert columns and general interest articles in each journal. Subscribe now and also receive the **HARDCORE COMPUTIST** eight times a year. Get in on an open discussion of every facet of the Apple including how to do and undo copy protection, make backups of locked software, use various copy utilities.

HARDCORE is available only through this ad. In all, receive both **CORE** and **HARDCORE** one issue a month for only $20.00.

SUBSCRIPTION

$20 U.S.A, $29 CANADA, $33 MEXICO, $42 ELSEWHERE

SAMPLES U.S.A. $5 ELSEWHERE $8

U.S. FUNDS ONLY
NO CREDIT CARDS

DEPT. N-1
P.O. BOX 44549 A
TACOMA, WA 98444
(206) 581-6038

NAME _____

ADDRESS _____ STATE _____

CITY _____

ZIP _____

Circle 16 on Reader Service Card

AMPERGRAPH

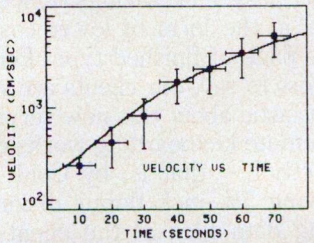
AMPERGRAPH is a powerful, easy-to-use relocatable graphics utility for the Apple II +/e. AMPERGRAPH adds twenty-two Applesoft commands that allow effortless generation of professional-looking plots of scientific or financial data. All of the necessary scaling and screen formatting is accomplished with just a few, simple Applesoft lines.

Unlike most other plotting systems for the Apple II which are stand-alone systems, the AMPERGRAPH utility provides extended BASIC graphics language macros that you can use directly in your own Applesoft programs. The additional commands are &SCALE, &LIMIT, &AXES, &GRID, &FRAME, &LOG X, &LOG Y, &LABEL AXES, &LABEL, &VLABEL, &CENTER LABEL, &CENTER VLABEL, &DRAW, &PENUP, &CROSS, &OPEN SQUARE, &CLOSED SQUARE, &OPEN CIRCLE, &CLOSED CIRCLE, &ERROR BARS, &DUMP (to dump the graph on a Silentype printer) and &*DUMP (to link with AMERDUMP, see below).

$45.00

SAMPLE AMPERGRAPH PROGRAM LISTING:

```
10 &SCALE, 0, 80, 80, 13000
15 LX$ = "TIME (SECONDS)":LY$ = "VELOCITY
   (CM/SEC)"
20 &LOG Y: &LABEL AXES, 10, 10
25 LABEL$ = "VELOCITY VS. TIME":&LABEL, 30,
   200
30 FOR T = 0 TO 80:&DRAW, T, 150 + T12:NEXT T
35 FOR T = 10 TO 70 STEP 10
40 &CLOSED SQUARE, T,
   (150 + T12)*(.8 + .4*RND(3) )
45 &ERROR BARS, 5, T12/2
50 NEXT T:&DUMP
```



AMPERDUMP

AMPERDUMP is a high-resolution graphics dump utility which can be used either in menu-driven mode, or directly from your Applesoft program, with, or without AMPERGRAPH. The following printers will work with AMPERDUMP: Epson MX-80, FX-80, MX-100; Apple DMP, NEC PC-8023A-C, C. ITOH 1550, 8510A/B, 8600. AMPERDUMP offers many features which are not available in other graphics dump routines:

* Horizontal magnifications: 3 with Epson printers (2.33 to 6.99 inches); 12 with all others (1.75 to 7.78 inches)
* Vertical magnifications: 9 with Epson printers (0.88 to 7.96 inches); 6 with all others (1.33 to 8.00 inches)
* Horizontal and vertical magnifications can be specified independently.
* Normal / Inverse dumps
* Fast
* Adjustable horizontal tab
* Easy to use
* Compatible with AMPERGRAPH
* Relocatable

$40.00

The AMPERGRAPH and AMPERDUMP graphics utilities require an Apple II +/e (or Apple II with language card). The AMPERDUMP utility requires one of the following interface cards: Epson, Apple, Grappler, Interactive Structures, Mountain Computer, Epson Type2, Tymac, or Microbuffer II.

AMPERGRAPH and AMPERDUMP are available from your dealer or order direct. Include $2.00 for shipping and handling; Wisconsin residents add 5% sales tax.

mad West SOFTWARE

121 N. Allen St.
Madison, WI 53705

MasterCard **608-238-4875** VISA

Circle 22 on Reader Service Card

TYPESET

"type" specifications as:

- Point-size(s) of type to be set.
- Type faces or styles (among over 400) desired by client.
- Line length of each type line.
- Any other special requirements.

Once the copy has been thusly encoded, it is "run" on Cold Type's digitized typesetting units, which output galley pages of finished, layout-ready type.

Due to stringent factors of time and budget, each weekly issue of *Wrap-up* must be typeset and keylined (prepared for the printers) in a single evening. This would be difficult to do (and rather costly, as well), using conventional all-in-the-shop typesetting procedures. But, thanks to remote typesetting, *Wrap-up* comes in way *under* budget — a welcome breath of fresh air to the state government!

"When you have typesetting *savings* to sell," notes Cold Type's President, "then you're really speaking the client's language, especially today. By expanding our operations, we're able to handle more remote-typesetting clients than ever before. We've even added more keyboarding personnel, for the more specialized typography jobs which must be keyboarded in the shop."

Cold Type appears to illustrate an economic fact of life which holds true, especially in a depressed economy. Namely: if a firm, by expanding, can place itself in a stronger position to help clients *save* money, then clearly the time to expand is now.

Says Simpson: "With the economy so sluggish, expanding our operation was really the *only* sensible route to take. *Just ask our clients!*"

A **Call -A.P.P.L.E.** PRODUCT REVIEW:

WINDOW

Michael Thyng/reviewer

WINDOW *$29.95*
$120/year

469 Pleasant St.
Watertown, MA 02172

WINDOW, as described by its creators, "makes Electro-Publishing history by boldly introducing an electronic magazine which exploits state of the art microprocessor/telecommunication technology in a magazine format." It is a diskette packaged in a corrugated cardboard mailer.

It was inevitable. With change occurring in our world so quickly, this product would have been produced by someone sooner or later. I'm really glad that the first ones to market such a product were Henry F. Olds Jr., and John Richards. They've done the job right.

I wanted to know how long it would take to "read" an electronic magazine, so I timed my way through it. If you spend longer playing the games or reading the reviews, of course you will spend longer. It took me 77 minutes. That is about as fast as you can go during the first "reading" — if you participate in the activities. But let me explain.

Part of WINDOW is text-on-the-screen. That's what I expected to see. Part of WINDOW is clever formatting of that text. Part of WINDOW is audio. It makes the most delightful and appropriate sounds to accompany the text. Part of WINDOW is interactive. You don't get to just sit back and watch it perform. All of WINDOW is educational. Go through it with me. This thing even has a table of contents. I just went down the list selecting one after the other. First is the Editorial. It was a little light for my tastes. I want to know that my editors are literate. Show me more words, but give me substance too. B in substance. D in length. A in showmanship.

It took me two minutes (first time blunders) to get to the next article. I won't have that problem with the next issue. The control commands are really easy to use, but the first timer will probably fall into the same trap I did.

The next feature is a review of an educational program from Beagle Bros. called WOWZO. *Can you picture an interactive review? You actually play a portion of the game.* Then WINDOW takes it away from you and describes the pluses and minuses. Now I want to run right out and buy WOWZO. Marketeers, listen to me. You are really missing a bet. This is a dynamic (and diabolical) way to sell a software product. I mean, you are actually putting the product in the customer's hands to show it. What an incredible sales tool. Contact Anthony Flesch or George Mercer at WINDOW and see if they take advertising. There wasn't any in Vol 1, Number 1. I hope that changes.

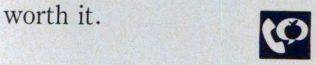
Next feature is Window on BASIC. I spent 29 minutes trying to figure out how to properly run this. This was their "weak link" in the chain of features. Yet, for every nasty thing I can think of to say about it, I have to tell you that I plan to use this program to demonstrate the logic of certain BASIC programs to my students. It is the kind of idea that causes people who see it to slap their foreheads and say, "now why didn't I think of that?" It's fully copiable for the WINDOW subscribers own use.

The feature article is a "How To" on improving your programming. I teach programming and I learned something. I believe it is quite a good article. After you have learned the programming practices, WINDOW gives you the program for your own use, and encourages you to make changes!

Next feature is called Digit Shuffle. This is an educational game that teaches you how to make estimates about the products of multiplication. I didn't spend as much time working with Digit Shuffle as I needed. But I can go back any time to practice. This feature looks like it was written by someone with some significant educational skills.

I didn't quite understand the purpose of Number Dropper. The Table of Contents says it is a sample of a new series of learning games for young children from Learningways. I tried it out on my five year old. He was pretty impressed.

Let me finish. Single copies of WINDOW sell for $29.95. Regular full year subscriptions cost $120.00, charter rates are $95.00. Is it worth the price? You get an educated editorial, three computer programs, programming instruction, a program review with hands on experience, and a preview of future products — again hands on. Associate Program Editor, Bill Crouch, tells me that Vol 1, Number 2 is already scheduled to have more, and future issues may go to two sides. Yeah. It's worth it. It's definitely worth it.



REQUEST FOR CHANGE OF ADDRESS

Dear A.P.P.L.E.,

I am moving and my present address is:

PRESENT:

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP CODE _____

EFFECTIVE DATE _____

NEW ADDRESS:

NAME _____

ADDRESS _____

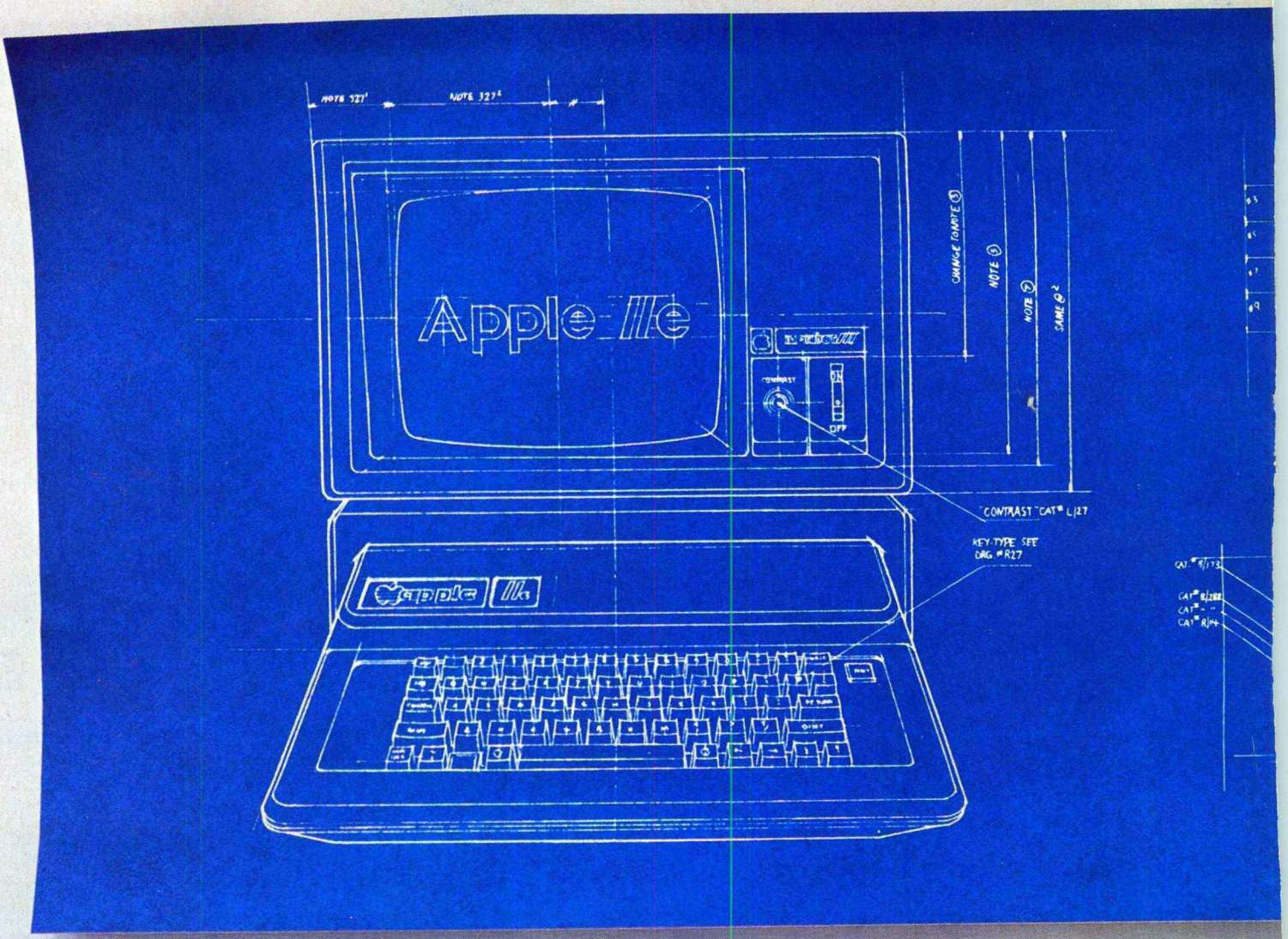
CITY _____

STATE _____ ZIP CODE _____

MEMBERSHIP NUMBER _____

*Please fill out and return this notice two weeks before moving.

It's the same old Apple II.



For years, people have been trying to build a better Apple® II.

It finally happened.

Meet the Apple IIe, an impressive new version of a most impressive machine.

The "e" means enhanced. Which means a bundle of new features:

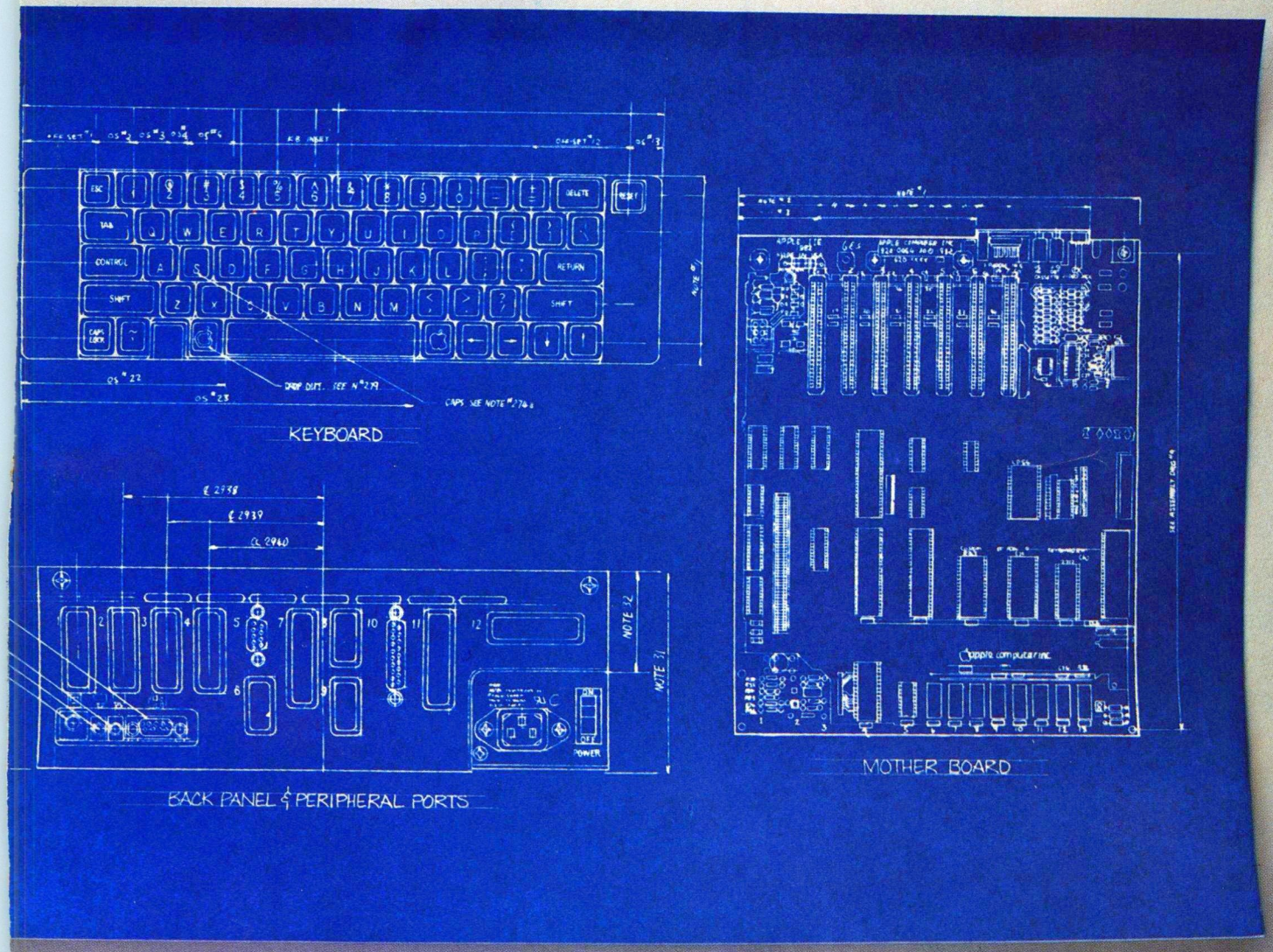
A standard memory of 64K (versus 48K) that's easily expandable. So you can create fatter files and crunch larger numbers of numbers.

A new, improved keyboard, with a complete set of ASCII standard characters. Plus full cursor controls, programmable function keys, and a rapid auto-repeat feature built into every key on the board.

Both upper and lower case characters. (And if you want to see more of them on the screen at one time, a low cost 80-column text card is available.)

Improved peripheral ports. Which make it a lot easier to connect and disconnect game controllers, printers and all those other wonderful things that go with an Apple Personal Computer.

Except for the front, back and inside.



Self-diagnostics. That's a special feature that makes it easy to give your computer a thorough check-up.

Plus an even more reliable design. Achieved by reducing the number of components—which is to say, the number of things that could go wrong.

And bear in mind, the IIe still has all those other virtues that made the Apple II so very popular. Including access to more accessories, peripheral devices and software than any other personal computer you can buy.

So visit any of our over 1300 authorized dealers, and see the newest Apple for yourself.

Like the original, it's rather extraordinary. But then some things never change.

apple®

The most personal computer.

Circle 99 on Reader Service Card

Call **(800) 538-9696** for the location of the authorized Apple dealer nearest you, or for information regarding corporate purchases through our National Account Program. In California **(800) 662-9238.** Or write Apple Computer Inc., Advertising and Promotion Dept., 20525 Mariani Ave., Cupertino, CA 95014. © 1983 Apple Computer Inc.

All Logical Paths Lead to

PDS DATADRIVE™ FOR YOUR APPLE II*

great price

15% greater storage

100% **Apple II*** compatible

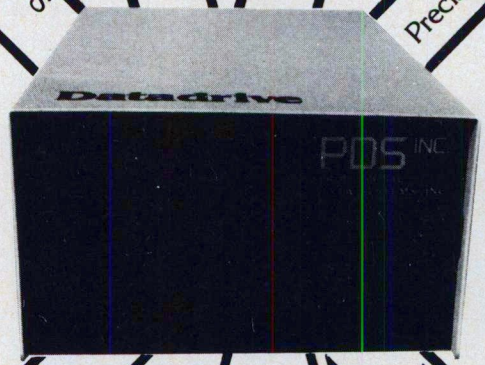
300% track-to-track speed

Outstanding value

Enhancer diskette and 40-track mode

Half-tracking, DOS PASCAL, CPM

Precision lead screen



PDS Universal, Inc.
2630 Walnut Ave., Suite G
Tustin, CA 92680

(714) 730-6772
Telex 18-3511
answer back CSMA

Disconnect Interface Board

Highest standards 8500 hours MTBF

Shugart** mechanism and read/write electronics

When Datadrive is not driving it uses no power

easy interconnect to other computers

super-silent operation long life

state-of-the-arts electro-mechanics

conserve power

*Apple II is a trademark of Apple Computers Inc.
**Shugart is a trademark of Shugart Corporation

Circle 8 on Reader Service Card

A Call -A.P.P.L.E. SOFTWARE REVIEW:

SPECIAL EFFECTS

Jeff Harper/reviewer

Special Effects $39.95
by Mark Pelczarski
Penguin Software
830 4th Ave.
Geneva, IL 60134
unprotected

THE Special Effects program by Penguin Software is an excellent enhancement to the Complete Graphics System II program, but it is also an outstanding picture graphics program in its own right. I was looking for a utility to take the work out of creating pictures on the Apple. My specific interest in graphics capabilities was in their application to interactive educational programs for preschool children. Graphics can make the difference between a program that will generate no interest from a normally inquisitive child to one which will spark interest from almost anyone. Graphics enhancements to almost any type of program can make a significant difference in its appeal to the user. While this article focuses on the features of the Special Effects software, this reviewer highly recommends the combination. Both Special Effects and the Complete Graphics System II provide all the functions almost any programer would require.

The Special Effects program presents the user with five modules for use in creating a multicolor two dimensional picture. These modules include the Brush module, Tricks module, Packing module, Viewstring module, and the Font/Picture conversion module. The program is menu-driven and each module is accessed by selecting an option from the main menu. Each module displays, in abbreviated form, the command options available in that module. From the main menu disk commands can be executed without exiting the program. A color bar option may be selected from the main menu to provide the user a reference for video display color adjustment. Six primary Apple colors are displayed.

The Painting module probably contains the most powerful tools available in the Special Effects program and accessed by selecting the appropriate option from the main menu. An abbreviated list of commands is displayed in the lower portion of the screen. A color graphics picture is created by moving a brush about the screen with the paddles or joystick (a tablet version of the software is also available). A "BRUSH UP" and "BRUSH DOWN" option allows the user to suspend painting, move the brush, and resume painting at a new picture location. Both the X and Y coordinates of the brush are displayed at the bottom of the screen, providing the user with the exact brush location at all times. Menu display may be toggled off to provide the entire screen for picture creation. The shape of the brush may first be selected from a variety of brush shapes and sizes displayed on the screen. These include circular, square, rectangular, and slanted bar shapes as well as some irregular blended shapes. Brushes may be changed at any point in the picture creating process.

Selecting a color for the brush is as easy as entering a number. The current brush number is displayed below the menu in the lower area of the screen. A color palette is available to provide a preview of the color number code selected. The colors include the Apple primary colors: black, white, orange, green, blue and violet. A total of 102 additional blended colors are achieved by combining these six primary colors.

While in the Painting module, the user has the convenient option to load or save drawings from disk without returning to the main menu. Current drawings are quickly saved on disk and previously created ones easily activated for additional modification. The Painting module also includes a magnification function. A selected portion of the picture can be magnified to either two or four times its original size.

The Tricks module provides the user with several functions to create mirror images and picture reversals. These can be useful when creating two dimensional, symmetrical objects or just for having some fun with the graphics. One half of the picture can be created first, then the other half generated conveniently by using one of the mirror functions. Picture flip options include mirror images from left to right, right to left, and top to bottom. All or half of the picture may be mirrored.

Several color options are also available in the Tricks module. Color swapping can be acheived and viewed instantaneously. One example would be an entire picture mirrored with orange colors swapped with blue and green swapped with violet. Various other color swapping options are included.

The Tricks module also provides the user a means of duplicating a selected portion of a picture from its original location to another location on the same picture. A rectangular area specified by the user defines the portion of the picture to be copied. A new location is defined and a copy of the graphics contained in the original area is generated.

A Packing module is included with the Special Effects program. Its function is to save disk space by reducing picture disk sector size. Most pictures created will require 34 sectors. In some cases these can be packed to as little as seven sectors. Also provided with the Packing module is the unpack function. Any modification to a picture using the Painting or Tricks module requires the file to be in the unpacked format. In addition to saving disk storage space, packed pictures may also be strung together using the viewstring module. A series of pictures can then be viewed in a predetermined sequential order.

SPECIAL EFFECTS

The Conversion module is a utility which provides the user with the capability of converting a picture from another format to the Penguin Software format or vice-versa. The most useful function of the module is probably the conversion of character sets from other software for use with the Complete Graphics System II.

The documentation for the Special Effects program is similar to that provided with the Complete Graphics System II. It is easily understood and very thorough. Included is a section in the appendix which summarizes all of the menu functions and lists the color number codes. A programmers notes section describes methods for including the routines in the users own programs. As with other software provided by Penguin, the utilities and routines included with Special Effects and the Complete Graphics System II are not copy-protected and can be easily included in the users' own programs.

ASSEMBLER/TEACHER

This program makes your APPLE* teach you assembler! We assume you know BASIC already, and now you'd like to learn APPLE* assembler. You need three things (two heavies and a friend) . . .

1. The APPLE II Reference Manual (heavy),
2. A good 6502 assembler manual (another heavy: ask for both at your APPLE* dealer), and
3. ASSEMBLER TEACHER, a friendly program that takes you right through the learning barrier!

ASSEMBLER TEACHER will soon have you eating those manuals for breakfast. And you'll never outgrow ASSEMBLER TEACHER, since it has tools an expert can use even more than a novice. Here's how ASSEMBLER TEACHER works:

- ASSEMBLER TEACHER prowls your APPLE*'s memory page by page, drawing a map of what it finds.
- ASSEMBLER TEACHER shows your CPU running in slow motion on your screen, so you can see exactly what the operations are doing to the registers and the stack (a great debugging tool).
- ASSEMBLER TEACHER exposes the mysteries of hexadecimal, two's complement, character and other data representations, instantly translating from any version to all others.
- ASSEMBLER TEACHER gives you the concepts you need to cope with assembler, using simple on-line lessons that grab you by the keyboard and make you play your way to knowledge.
- ASSEMBLER TEACHER is written by John Fairfield, a Ph.D. in Computer Science who regularly teaches assembler. A RAM Mini-Assembler is included.

"Rating: A . . . a great help to anyone interested in learning machine language programming . . . and the price is right."
Peelings II (Jan. '82)

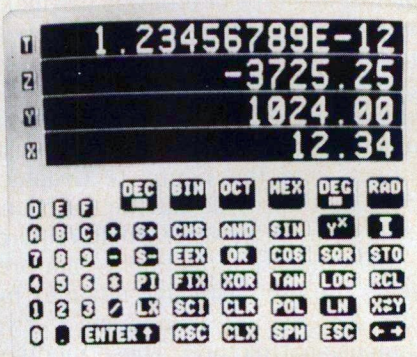
Ask for
ASSEMBLER TEACHER
at your dealer, or order from:
ASSEMBLER TEACHER
Computer Works, Inc.
P.O. Box 1111, Harrisonburg, VA 22801
(703) 434-1120

$44.95 MC VISA DOS 3.3

*Registered trademark of Apple Computer, Inc.

Circle 4 on Reader Service Card

FOR THE APPLE II®



HI-RES VERSATILE CALCULATOR™

Powerful tool for Teachers, Programmers, Engineers

- **Over 50 functions** - arithmetic, scientific, engineering, statistical, ASCII codes, & more
- **Comprehensive User's Manual** - features large Tutorial section. Comes in 3-ring binder.
- **Operates in 4 Bases:** Decimal, Octal, Hex, Binary
- **Instantaneous Results using RPN Operation**
- **Highly Visual Display** in color or B/W
- Features Internal Text Reference Guide
- Demo Program included

Requires 48K, disk drive, ROM Applesoft
$59.95 + $2.00 handling (FL res. add 5%)

Check your local dealer or CONTACT:

TACKABERRY SOFTWARE

TACKABERRY SOFTWARE
P.O. Box 2857
Ormond Beach, FL 32074
Phone: (904) 677-2054

® Registered Trademark of Apple Computer, Inc.

Circle 36 on Reader Service Card

APPLEMASH:

BORED WITH BULLETINS?

Michael Thyng/Call -A.P.P.L.E. staff writer

TODAY I'd like to do a follow-up article to one that Jeff Finn wrote in the September 1979 *Call -A.P.P.L.E.* magazine. The subject is Bulletin Boards in general and the *A.P.P.L.E. Crate* and *Apple][Crate* in particular.

Do you know what a Community Bulletin Board is? It is a computer program controlled data base running on a dedicated computer which is connected to a telephone line. In our case this means an Apple][computer is constantly (during hours of operation) monitoring the phone line, waiting for a call. When a call is received, the program logs it in and starts feeding information and data to the caller.

Most bulletin boards have some sort of verification procedure to detect the legitimate users from the troublemakers and clowns. This procedure usually takes the form of a password that the computer assigns to the user. The user then signs into the bulletin board system, BBS, using the password.

The computer then checks the password against its master list and approves or rejects the caller. If the user's password proves to be valid, the user is free to read most messages and leave messages for other legitimate users. If the user's password is invalid, then the computer hangs up. The two BBS's with which I am most familiar are A.P.P.L.E. Crate, sponsored by A.P.P.L.E., and the Apple][Crate, a privately run BBS. Both systems are open to the general public. Other BBS's may have a charge for their services. Two of the more well known are Compuserve and The Source.* Of course, they have programs available for downloading, while the "Crates" do not. But then the "Crates" are free!

Why would someone want to call a BBS. There are many reasons. Let me just touch on what I consider to be the most practical.

Perhaps you've heard of the concept of electronic mail. What is that?

Well, it's mail that is electronically transferred instead of stamp-envelope-and-paper transferred. That means you can type (or maybe I should say key in) your message into memory in your Apple][(or memory retentious terminal).

Then you would dial up the BBS which would allow you to store that message.

After your password was verified and you selected the option to leave a message, you would direct your Apple][(or memory retentious terminal) to transmit your message to the BBS. The BBS would accept your message.

Next you would either sign off or scan the BBS for any messages that were left for you. The "Crates" automatically tell callers which messages have been left for them. If you don't have a data transmission program for your Apple][(or *III*) or your terminal, you can still leave a message on the "Crates." The "Crates" have, built into the software that runs them, a text-message editor that will let you call in and then type your message.

On other BBS's you can leave programs or load programs from the BBS to your computer. Sometimes you can run a computer program on the BBS. One of the most useful things micro owners do with BBS systems is record data on the micros a little at a time and store it in large capacity files at the BBS facility. Then when all the data is in, run a big number crunching program that the host computer at the BBS site has. After the program run, the micro can receive just the results or access parts of the updated data that the host computer is storing for the user. This makes the best use of the micro and best use of the "big" machine. On the community bulletin board systems, the users generally ask for and give help in such things as programming tips, hardware tips on maintenance and recommendations on types of printers to avoid and what who thinks about them. Some BBS's specialize in certain topics like horse-racing, helping the handicapped, adventure game teleplaying, and many others.

At the very least, it puts you in contact with a network of computer users who share a common interest, the Apple computer. Better, you can tap some of these folks for information. At best, you can meet users of the BBS who share the specific area of use to which you are trying to put your Apple.

How do you get to tap into such a thing as a BBS?

Well, you need a modem and/or an acoustic coupler. A modem is an electronic device that translates incoming tones from the phone lines into serial digital data for your computer and serial data from your computer into outgoing tones for the phone lines. Actually, the modem doesn't know whether the tones are incoming or outgoing, but translates tones of two different frequencies into the 0's and 1's of serial data that your Apple treats as incoming data. The modem translates serial data sent to it into tones of two different frequencies. That way there is no danger of mixup between incoming and outgoing data. Modem is an acronym which stands for modulator-demodulator.

A modem will typically be an electronic device which plugs into one of your Apple slots and also into the phone jack. It bypasses the telephone.

An acoustic coupler is a cradle which holds a telephone handset and does the job of translating the audio signals to digital and the digital signals to audio through analog to digital and digital to analog means respectively.

While both do the job, I would prefer the direct connect modem. It connects directly to the phone jack (either with or without your telephone) and hence, should have more immunity to noise.

* *See box on next page for more information.*

BORED WITH BULLETINS?

What do they cost? They should cost you between $139 and $629. The wide variance is because of differences in brand, quality, and baud rate. Baud rate is a measure of the speed that the data travels over the phone lines. I think most micro owners transmit data at a baud rate of 300. This translates to about 30 characters per second. The phone lines will work at rates of up to 9600 baud. But most people find that 1200 baud is a realistic upper limit on the amount of data that can reliably be transmitted with today's technology. (Modem freaks, correct me if you have other information.) I think that your best buy for the money is a modem that lets you work reliably at 300 baud.

1200 is nice, and if you can get it, go for it. But if you go for 1200 baud make sure it will also work at the slower 300 baud. I tell you this because I think you will have access to more hobbyists at 300 baud than at 1200.

Now, how do you get to call the "Crates"? A.P.P.L.E. Crate is available almost 24 hours a day at (206) 935-9119. Apple][Crate is available at (206) 525-5410 from 8 p.m. to 8 a.m. weekdays and almost 24 hours on the weekend. Remember we're talking Pacific Time Zone. After your first call, *if* you ask to be assigned a password, you will have to wait 24 to 48 hours for the SYSOP (system operator) to validate your I.D. on the A.P.P.L.E. Crate. The SYSOP on Apple][Crate will validate you promptly in from three minutes to two weeks after you apply. Generally it works out that the wait time is inversely proportional to your need to leave a message.

Who can use these systems? Actually anyone who can convince the SYSOP to let them on one of the systems is eligible to use the system. Since the data is transferred back and forth in ASCII format — I don't want to explain that this time — anyone whose terminal or computer-modem can send and receive data in ASCII format is welcome to use these BBS's. So, your friends with TRS-80's can call in and so can the Vic-20 people. Even people who have the, oh what's its name... they made big computers for years and years... you know, it's right on the tip of my tongue... they named the cards after them ... oh, yes, IBM. Sure, if anybody makes a modem for them, they are welcome to call in too.

A.P.P.L.E. on the SOURCE

A special section of The Source is open only to A.P.P.L.E. members who may download Call - A.P.P.L.E. programs. To access, type:

CALL-APPLE from Source command mode. To register, send mail to ST1570 with your name, membership number, expiration date and Source account number.

StarLogic Announces Savings on Apple II Compatible Drives

Includes drive, cable, cabinet and standard warranty which includes 90 days parts and labor

5¼" standard disk drive	$205.00
ThinLine half-height disk drive	$185.00
Dual ThinLine drives	$335.00

(Also compatible with Franklin ACE)

TELEPHONE ORDERS ONLY
MASTERCARD, VISA, CASHIER'S CHECK COD ORDERS ACCEPTED
(213) 883-0587

StarLogic

Apple is a registered trademark of Apple Computer, Inc.

ThinLine is a registered trademark of Tandon Corporation

Circle 34 on Reader Service Card

DOUBLE MONITORS FOR THE APPLE][

Greg DeWilde

"THERE are two Monitor ROMs available for the Apple][. The two Monitors are identical for most functions. They differ only in certain features." This quote, taken from the introduction to *Apple][Monitors Peeled* from Apple Computer, is what this article is all about. First we will look at the important differences between the two Monitors, then present a circuit to allow both Monitors to reside inside the Apple at the same time.

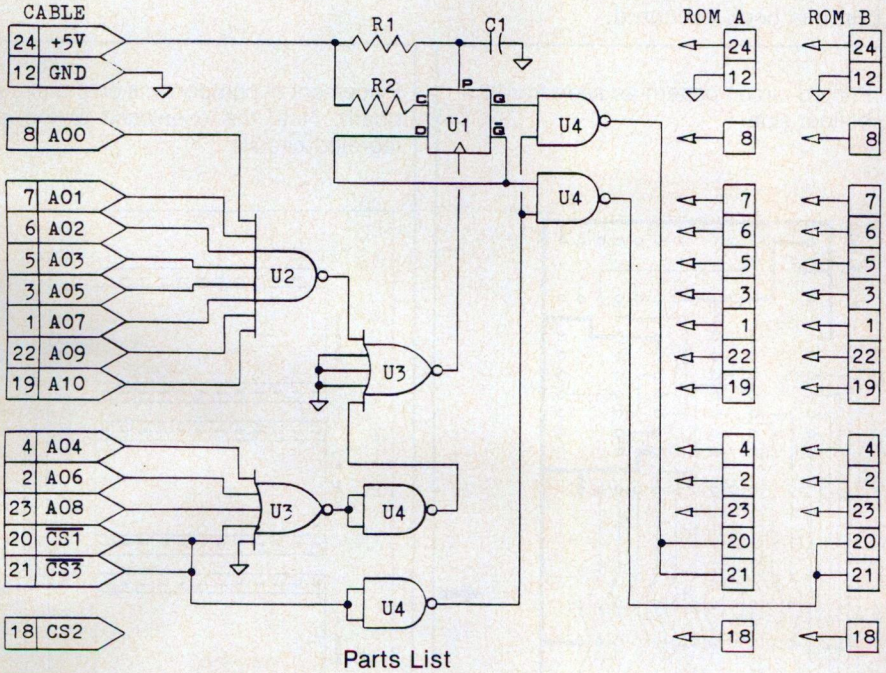
Early Apple][s were shipped with the original Monitor, known as the Old Monitor. In 1979 the new version of the Monitor, the Autostart, became available, first as a replacement for the Old Monitor, then as the supplied Monitor in the Apple][Plus.

It is easy to determine which Monitor is in your Apple. On a power-up or reset, if the legend APPLE][appears at the top of the screen the Autostart Monitor is installed. If the Apple comes up with the * prompt, it contains the Old Monitor. Another method is to PEEK at location -1281 ($FAFF). The Autostart Monitor contains 00 and the Old Monitor 01.

Differences

The most obvious difference between the two Monitors is that the Autostart will operate the Apple][like a turnkey system. That is, it will automatically decide what to do on power-up or reset, rather than asking the operator for a monitor command like [Ctrl-B] to initialize BASIC. If a disk controller card is plugged into any slot, the Autostart will load DOS and attempt to execute the HELLO program on the disk. If no disk controller card is found, the Autostart initializes the language in ROM. This may be the ROMs on the motherboard or on a ROM card installed in slot 0, depending on the position of the switch on the ROM card.

Figure 1. Schematic and parts lists for the DBLMON adaptor board for the Apple][. With the cable plugged into socket F8 on the Apple motherboard, the circuit will switch between the two Monitors installed in the ROM A and B sockets whenever a PEEK at address $FEAF (-337) takes place. On power-up, the circuit will select the ROM A position.

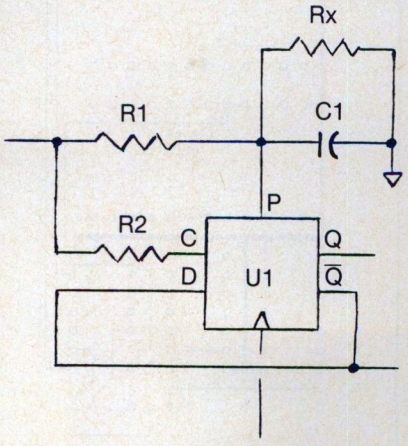


Parts List

QTY	ID	Description
1	C1	.033uF dis capacitor
1	R1	470k 1/4W resistor
1	R2	4.7k 1/4W resistor
1	U1	74LS74
1	U2	74LS30
1	U3	74LS260
1	U4	74LS00
1		7"-24 conductor cable
2		24 pin dip headers
3		24 pin dip sockets
4		14 pin dip sockets

Figure 1a.
Since I wrote this article, one modification to the circuit was necessary to prevent false selection on quick power off/on occurences. A 1Mohm resistor in parallel with C1 solved the problem (see dwg below).

Add Resistor Rx, 1Mohm
¼ or ⅛ WATT Solder to foil side of board, across C1 leads.



DOUBLE MONITORS FOR THE APPLE][

Another important difference is the cursor movement commands. The Old Monitor required the key ESC followed by an A, B, C or D to move the cursor right, left, down and up respectively. This sequence must be keyed in for easy movement of the cursor. With the Autostart Monitor, after pressing the ESC key the key I, J, K or M may be pressed in any combination to move the cursor without pressing the ESC key again. This is now commonly referred to as the "diamond cursor pattern".

When LISTing files with the Old Monitor, the entire program scrolls by unless a range of line numbers is specified first. The Autostart Monitor includes the stop-list feature [Ctrl-S]. By pressing [Ctrl-S] at any time during the listing of a program, or almost any other screen output, the display is halted until another key is pressed.

From the above features, it would seem that the Autostart Monitor is a much better choice and that anyone with an Apple that still contained an Old Monitor would want to replace it as soon as possible. There are, however, two important functions in the Old Monitor that were eliminated in the Autostart to make room for the new cursor commands and the stop-list feature.

The STEP and TRACE functions are especially useful to anyone who programs in assembly language or to those who include assembly routines in their BASIC, Pascal or other high level language programs.

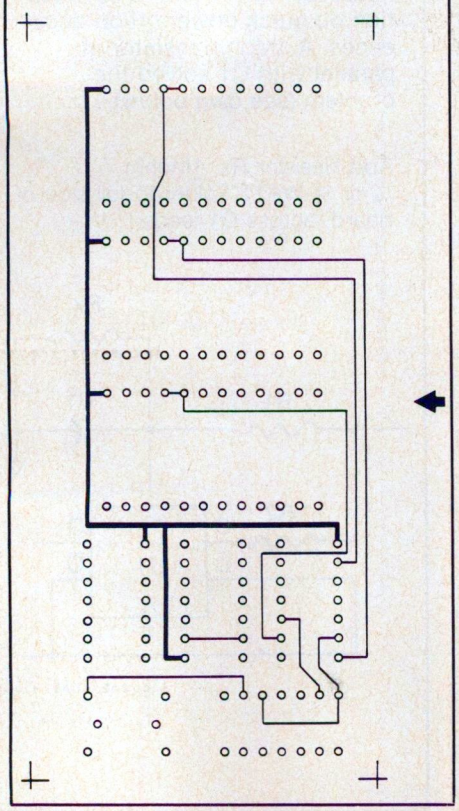
The STEP functions allows the programmer to single step through the machine code instructions. It first decodes the instruction into its disassembled form and then displays it. Next the instruction is executed. Finally, the contents of the 6502 CPU's registers are displayed for examination. The monitor then waits for the programmer to ask for the next instruction to be executed.

The TRACE function executes an entire assembly routine or program, displaying the assembled version of the instructions and the contents of the CPU registers after each instruction has been executed.

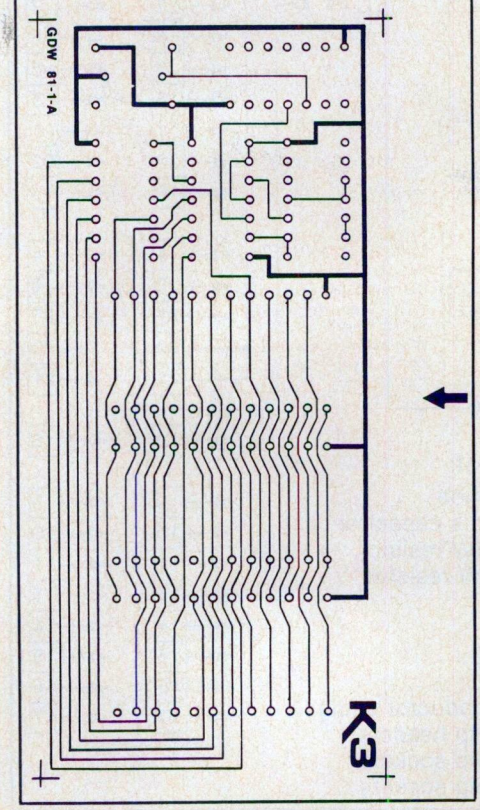
Both of these commands are very useful in debugging assembly language routines or in understanding the operation of a routine already contained in a program or even in the Monitor itself. There are assembly language development tools available that could be used in place of the STEP and TRACE functions, but we will assume that the addition of these particular functions is desired.

Perhaps the easiest way to have the features of both the Old and Autostart Monitors is to have both Monitors available in the Apple. Except for the special features described above, the two Monitors are essentially identical. Therefore, it is necessary to have only one of the Monitors active at a time, depending on the feature required. The DBLMON adaptor board to be described below allows both Monitors to occupy the same memory space, $F800-$FFFF.

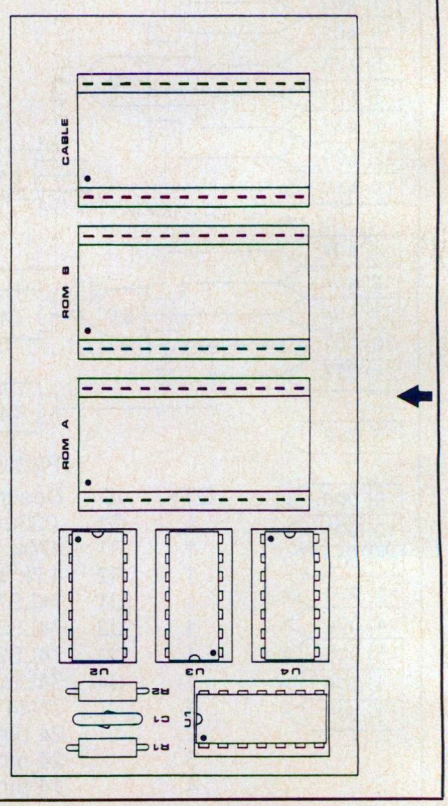
Printed circuit pattern for the DBLMON board as seen from the component side.



Printed circuit pattern as seen from the bottom side.



Placement of components on the circuit board. Note the orientation of the integrated circuits.



DBLMON Adaptor

Instead of occupying an expansion slot, the DBLMON adaptor board plugs into the F8 ROM socket on the Apple motherboard in place of the existing Monitor. On a power-up, the DBLMON selects the Monitor installed in the ROM A position. Under software control, the user can then select the Monitor in either the ROM A or ROM B position, even from inside a program. A reset command will not alter the Monitor selected, only another power-up will default to the ROM A Monitor. To ensure that the selected Monitor will not change except when requested, the address used to activate the switching circuitry in Figure 1 must not be an address that will be accessed by the computer for any reason. Both the Old and Autostart Monitors contain a NOP (no operation) instruction at address $FEAF that is between two routines. This address should never appear on the address bus during normal operation of the computer. By PEEKing at this location (-337), the DBLMON will alternately select the ROM A or B position. Which Monitor is selected can be determined by PEEKing at location -1281 as described earlier.

Figure 2a and 2b show the printed circuit pattern for the DBLMON from the component side and the bottom side, respectively. The component placement is shown in Figure 3. Note the orientation of pin 1 of the integrated circuits. Figure 4 is the ribbon cable to connect the DBLMON board to socket F8 in the Apple. A standard 24 pin jumper cable may be used, but inverting one of the headers as shown makes for easier installation. A minimum of six inches of cable allows the DBLMON to be installed on top of or on the side of the Apple's power supply with double-sided tape to insulate the DBLMON from the power supply's case.

Other Uses

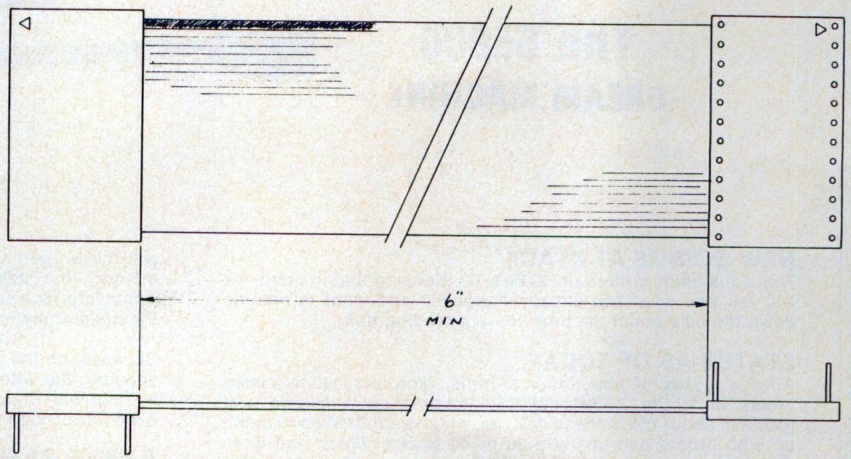
The DBLMON board may be used to double the capacity of any of the ROM sockets in the same fashion as for F8. Caution must be used as to the contents of the ROM at the switching addresses.

The switching addresses for the motherboard sockets are as follows:

F8	-	$FEAF
F0	-	$F6AF
E8	-	$EEAF
E0	-	$E6AF
D8	-	$DEAF
D0	-	$D6AF

Instead of containing the two Monitors supplied by Apple, the more experienced user may wish to program a custom Monitor or utility routines into a ROM and then use it in conjunction with one of the Monitors.

Figure 4. The cable to connect the DBLMON board to the Apple F8 socket. A standard dip jumper could be used, but the inverted header on one end eases installation.



BYTES & PIECES

Extra Special!

d Base II [Req. - Z-80]

$449.46

$1.00 credit for all phone orders over $100.00

	List	Ours
Bag of Tricks	39.95	29.56
Bank Street Writer	69.95	51.76
Bug Byter	47.50	35.16
Graforth	75.00	55.49
Graphics Magician	59.95	44.36
List Handler	89.95	66.56
Logo (Krell)	89.95	76.46
Nibbles Away II	69.95	51.76
Quick Code (req. Z-80)	295.00	218.19
Replay II	150.00	129.49
16 K Card (Microsoft)	99.95	77.96
Step by Step II	89.95	66.56
System Saver fan	89.00	76.46
Tasc Compiler	175.00	129.49
Word Handler	199.00	147.26
Wizardry	49.95	36.96
Zoom Graphics	49.95	36.96
SPECIALS		
Multiplan (CPM or DOS)	275.00	198.46
Wildcard	139.95	111.19
Home Accountant	74.95	52.46
T.G. Trak Ball	64.95	44.96
Microbe	44.95	31.96
Ultima II	59.95	41.96
Pinball Construction Set	39.95	28.56

C.O.D. • Money Orders • Certified Checks • Personal Checks Allow 2 Weeks • N.Y.S. Res. Add Sales Tax
U.S.-Orders Under $150. Add $2.00 P&H
All Canadian, U.S. Funds $3.00 P&H
Foreign, Charges Only, Min. P&H $6.00

• SEND FOR FREE PRICE LIST #900 •
Source TCP637

BYTES & PIECES (516) 751-2535

Box 525 Dept. C • E. Setauket, N.Y. 11733

Circle 6 on Reader Service Card

« DAJAX »

Save Time Anytime.

Software For Your Apple
By Direct
Modem Downloads

Requirements:
Apple II/II+
48 K or more
DOS 3.3
D. C. Hays Micro Modem II

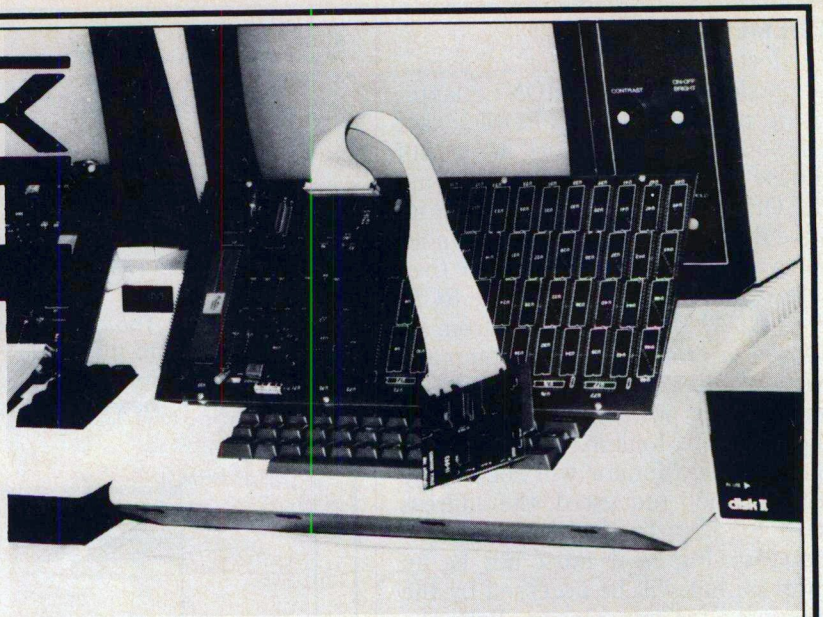
« DAJAX »

CALL 24 HOURS A DAY.
(404) 736-3529

Visa And MasterCard
Only Accepted

Circle 67 on Reader Service Card

DTACK 10



The 68000 DREAM MACHINE

NEW DOINGS AT DTACK

This ad is being written on 10 Feb '83. Keeping that in mind, we will tell you what has already happened and what is coming down the pike about the time you are reading this.

STATUS AS OF TODAY:

After a hiatus of about four months, Motorola has resumed production of the 12.5MHz 68000. In the past four weeks we have received about 150 of these devices, which permitted us to catch up with orders and upgrade 30 or 40 boards which had been delivered as, temporarily, 8MHz units. We actually have a surplus of L12s. Incidentally, most of our sales these days are 12.5MHz boards.

We mailed our newsletter #17 this past week, so we have over 300 pages of newsletter in print now. The more recent issues have been longer. For instance, #15 ran 32 pages and included the source code to a complete double precision floating point package, over 900 lines of code. A matrix inversion routine using the code was included. Issue #16 ran 30 pages and included a complete transcendental package using that double precision package, 845 lines of source code. In addition to the source code itself, we included 5 pages of explanations of how the transcendental functions work, what a Chebyshev approximation is and stuff like that.

For $77 an outfit in West Germany will sell you a chess program which requires a 60K DTACK board. This program is intended for serious tournament work. If you don't like games you can buy the Forth language for our DTACK board for $30 (add $15 more for assembly listings).

HARDWARE DOINGS:

We have designed a more complex Apple II interface card to allow transfers of blocks of memory between the 68000 and the 6502 at DMA rates. This allows, for instance, an 8K HIRES graphics page to be moved from the 68000 to the Apple II in less than 12 ms. That's less than the time for one screen update at 60Hz! Various sized blocks can be transferred and the transfer is bi-directional. The trickiest part was assuring that the transfer would **never** stomp on a disk read, or (horrors!) a disk **write.**

The prototype is fully functional and the layout has been farmed out to a consultant so that we can concentrate our efforts on the next project after that. By the time you read this, it should be possible to buy that special interface board for about $95 to $125.

The next project? Oh, yes. For those of you who don't need to run at 12.5MHz with no wait states, we are in the final design stages for our dynamic RAM version of the DTACK GROUNDED board. This board will have one wait state (DTACK is **not** grounded!) That board should be available about 30 days after the first appearance of this ad.

Some details: the board will have at least 508K DRAM, which is a smidgen (= 4K) less than half a megabyte. An expansion board will follow, permitting a megabyte of nice linearly addressable memory (the 68000 does **not** use memory segmentation). The Apple interface and the expansion connector will be **identical** to the present static RAM board.

Because of the wait state and a hardware/software refresh scheme, the effective clock rate will be equivalent to 10MHz without wait states. That's **twice** as fast as another, much more expensive 68000 machine which was recently announced!

A 68000 BASIC IS COMING!

Perhaps the best news of all is that Phase Zero Ltd., the outfit which sells the 68000 cross-assembler we have standardized on, is seriously working on a 68000 BASIC which will work with our DTACK board. Phase Zero is thoroughly familiar with Applesoft and intends that their 68000 BASIC be largely compatible with Applesoft.

SHOULD YOU BUY OUR BOARD?

We have received several letters criticizing our advertisements. Apparently we are supposed to say something like "everybody come buy a board right now". That isn't right, because our board is **not** for the mass marketplace at this time. It is (or they are) for the serious hacker, engineer or business type who wants to get a head start over the crowd by learning **early on** how the 68000 works and how best to apply it.Therefore, we describe our board as a development tool. We think that's accurate.

The guy who is writing this ad has written commercial assembly code for the Intel 4004 and 4040, for the Mos Technology 6502 and, for the past 20 months, for the 68000. If he was looking for a job, he could not find one programming the 4004/4040 today. He could not find a **good** job today programming the 6502. A word to the wise! Tempus is fugiting and the clock **don't** run backwards!

All is not lost. If you never programmed the Intel 4004, you probably don't know that the 6502 is much easier to program although it is also a much more complex part. Please accept our word that the 68000 is about four times easier to program in assembly language than the 6502 even though it is a much more complex part. Most people think that more **complex** equals more **difficult.** The reverse is true when it comes to programming microprocessors.

And we help you get going by providing 68000 source code on unlocked, easily copyable disks. Lots of it. Including the double precision floating point and transcendental package we mentioned earlier.

Perhaps you are a person who **should** buy our board?

DIGITAL ACOUSTICS
1415 E. McFadden, Ste. F
Santa Ana, CA 92705
(714) 835-4884

Circle 9 on Reader Service Card

Apple, Applesoft and Apple II are trademarks of Apple Computer Company.

SPELLING CORRECTOR

Wesley R. Felty

IF you already have a *Spelling Checker* program such as The Apple Speller, then you know how easy it is to check the spelling in your text files. But then, what about *correcting the spelling?* Most spelling checkers mark the misspelled work, but leave it up to you to find the marked words and correct them. Wouldn't it be nice to have a program that could quickly scan your file for a special symbol, ask for the correct spelling, and replace the marked word with the corrected spelling? Then you need *Spelling Corrector!* No more using a search routine and having to retype the whole line. No more sitting and waiting. Spelling Corrector is *fast*.

How Spelling Checkers Work

Most spelling checkers operate about the same basic way. First they load their dictionary into memory, and then they scan your text comparing your words to theirs. If one of your words is *not* on their dictionary list, then it shows you the word and asks whether to ignore it, add it to the dictionary disk, or to mark it as misspelled. It is marked by replacing the last letter with a symbol that most likely is not in your text, usually "^". You are *not* told where the errors occurred but are expected to scan the text later to find and replace the marked words.

One good thing about some of these programs like "Apple Speller" (new name "Sensible Speller") is that you can add your own words and thereby customize your dictionary. You can add names and places that aren't in more general dictionaries and add your own custom vocabulary (such as LDA, ORA, RTS, MAXFILES, etc.)

One bad thing is that the dictionary disks can't tell if you *have a correctly spelled, wrong word.* It would find nothing wrong with "They took *there* money." or "They went *their*."

Spelling Corrector is part BASIC and part machine language. The search for the error is done in machine language since the BASIC MID$() function is very slow. The removal of the misspelled word, inputting of the new spelling, and insertion of the corrected spelling is all done in BASIC.

How It Works

The main work of *Spelling Corrector* is done by the machine language program, "CORRECTOR". The BASIC program loads the file and then uses a normal "PRINT" command to print it to the screen. But "CORRECTOR" sets up an "output filter." It intercepts the output and searches for the spelling checker symbol (stored as "ERRTOK," *ERR*or *TOK*en, in line 35.) The value stored there, $DE, is for the Apple Speller's normal error symbol "^". If the user has picked the option to change this symbol while the BASIC program is running, line 160 POKEs the token to search for there.

As "CORRECTOR" prints out the intercepted line, it stores the number of the character being printed at "TSTPOS," *TeST POS*ition, at line 33. If it detects the error token, line 17, it loads the position into "ERRPOS", *ERR*or *POS*ition, line 20; rings the bell, line 25; and continues on until the end of the line. More than one error can be detected for each line. "ERRPOS" will hold and display the position of the *last error* or "0" if there were no errors on the line.

"CORRECTOR" also checks to see if the user has indicated that he/she has a lower case chip and prints to the screen accordingly. In the BASIC program, line 60 should be change to "LC = 1" if the user has a lower case chip. The "0" for no chip or "1" for a L.C. chip present is POKEd by BASIC lines 80 or 100 into the machine language LCFLAG, line 36. If a L.C. chip is indicated, the machine language program outputs directly to the screen the upper or lower case character (lines 42 and 43). If there is no L.C. chip, the program branches to "LC0," line 44, where it converts lower case characters to upper case (not in memory . . .only to the screen display), and converts upper case letters to inverse display (lines 53 and 54).

When each text line has been printed to the screen through the "output filter" in BASIC line 240, the BASIC program checks to see if there were no errors. Is "ERRPOS" equal to "0" or not equal to "0" (giving the position of the last error? This is done with the "PEEK (821) <>0 in line 260. The POKEs in lines 240 and 260 write directly to the top of the screen the number of the line being tested and the position of the last error in the line respectively. If no error is detected, line 280 sends in the next line to scan.

If an error is detected, the program jumps to line 300. The two "CALLs" in line 300 are how you turn off the output filter. (The CALL in line 180 is what turned it on). Lines 340 through 460 search backwards in the line for a space to cut out the misspelled word. They then ask for the correct spelling and replace the misspelled word with the new spelling. Three routines are needed to concatenate the strings depending on whether the error symbol is the last character of a line, middle character, or the first character of a line (a misspelled single letter word). This working from the end of a line back toward the beginning is done due to the fact that the spelling checker marks the *last* character of any misspelled words.

After the error is corrected, the line is displayed to show the correction by line 200, and then the line is checked again to correct any additional errors. Notice that output filter is turned off (CALL -365: CALL 1002) whenever a "normal" message is written to the screen. The filter is only on (CALL 768) when text is being written to the screen. Since the Apple writes upper case letters to the screen, they would be inverse if the filter were on.

SPELLING CORRECTOR

Lines 780 through 940 are the standard lines to READ a text file into memory from disc and lines 1020 through 1120 are the standard lines to SAVE a text file to disc.

Lines To Change

Lines 840 and 1040 most likely need changing for each person's text files. In my files, I save at the beginning of each the following parameters of the file. I = number of lines in the file. SL = length of the printout line. TB = number of spaces to tab. CL = number of columns to print out in. And CI = the print size (character/inch). Change these lines to READ or SAVE whatever data your own text files uses.

Using The Spelling Corrector

The spelling corrector is very easy to use. Just be sure to have "SPELLING CORRECTOR" and "CORRECTOR" on the same disk and "RUN SPELLING CORRECTOR." It will automatically load "CORRECTOR." You may now switch to the disk that has your checked files. Just follow the prompts.

After entering the symbol used by your spelling checker, your text will start to fly by. (This program can scan 300–80 character lines in 47 seconds). If any errors are detected, you will hear one beep for each error. The error symbol will display flashing and the top of the screen will tell you the position in the line of the last error. Be sure to enter the correct spelling for *the last error* first (if more than one error). The *last* misspelled word will be removed and replaced by the correct spelling that you enter. You will get used to telling how many errors each line has by the number of beeps. After a word is replaced, the line will be displayed to show the correction. If you enter a correction wrong, just write down the line number and use your normal editing routine. If you make an error while entering the word, just use the left arrow or [Ctrl-X] to abort it.

Upper and Lower Case

SPELLING CORRECTOR will display upper and lower case letters whether you have a chip or not. It is set for no lower case chip installed. If you have a lower case chip or board, just change line 60 to "LC = 1." If you don't have lower case on your computer, CORRECTOR will convert lower case letters to normal upper case and upper case letters to Inverse letters. When entering a spelling correction, you will need to enter upper or lower case letters to replace those in the text. The normal mode is lower case since most text is in lower case. To enter one letter capitalized, enter "ESC" followed by the letter. To shift-lock into upper or lower case, enter [Ctrl-S]. Neither the "ESC" or the [Ctrl-S] will be entered into your text. Since the SPELLING CORRECTOR removes the marked word all of the way from the mark forward to the previous space, be sure to include everything.

```
]LIST

10  REM

SPELLING CORRECTOR : WES FELTY

 CALL -A.P.P.L.E. : MAY, 1983

20  DIM L$(500)
40 D$ =  CHR$ (13) +  CHR$ (4): PRINT
     D$;"BLOAD CORRECTOR"
60 LC = 0: REM  LC FLAG...LC=0 IF
      NO LOWER CASE CHIP, LC=1 IF
      L.C. CHIP PRESENT
80  IF LC = 0 THEN  POKE 823,0
100  IF LC = 1 THEN  POKE 823,1
110  TEXT : HOME
120  PRINT "ENTER THE FULL NAME O
     F THE TEXT FILE TO CORRECT
      ": PRINT : PRINT "HIT [RETU
     RN] FOR A CATALOG": PRINT : INPUT
     NM$: PRINT : IF NM$ = "" THEN
      PRINT  CHR$ (4);"CATALOG": GOTO
     120
130  GOSUB 780
140  TEXT : HOME : PRINT "ENTER T
     HE SPELLING CHECKER SYMBOL !
     ": PRINT : PRINT "HIT [RETUR
     N] FOR ^": GET SE$: IF SE$ =
      CHR$ (13) THEN SE$ =  CHR$
     (94)
150  HOME : HTAB 4: PRINT "ERROR
     AT POSITION           LINE": PRINT
     : PRINT "HIT [ESC] TO CAPITA
     LIZE.....[CTRL-S]  TO TOGG
     LE SHIFT LOCK": PRINT : POKE
     34, PEEK (37)
160  POKE 822, ASC (SE$) + 128
180  CALL 768:X = 1: PRINT : POKE
     820,1: POKE 821,0: GOTO 220
200  IF Y (  ) 0 THEN Y = 0: POKE
     821,0: PRINT : PRINT L$(X): CALL
      - 365: CALL 1002: PRINT : PRINT
     : PRINT "HIT ANY KEY TO CONT
     INUE": GET A$: PRINT : CALL
     768
220  IF X ) I THEN  CALL  - 365: CALL
     1002: PRINT : PRINT "DONE": GOTO
     960
240  PRINT L$(X): PRINT : POKE 10
     60, INT (X / 100) + 176: POKE
     1061, INT ((X -  INT (X / 10
     0) * 100) / 10) + 176: POKE
     1062,(X -  INT (X / 10) * 10
     ) + 176
260  IF  PEEK (821) (  ) 0 THEN Y
      =  PEEK (821): POKE 1046, INT
     (Y / 100) + 176: POKE 1047, INT
     ((Y -  INT (Y / 100) * 100) /
     10) + 176: POKE 1048,(Y -  INT
     (Y / 10) * 10) + 176: GOTO 3
     00
280 X = X + 1: GOTO 220
300  CALL  - 365: CALL 1002
320  PRINT :WK$ = L$(X)
340  IF  LEN (WK$) ) Y THEN RT$ =
      RIGHT$ (WK$, LEN (WK$) - Y)
     : FOR S = Y TO 1 STEP  - 1: IF
      MID$ (WK$,S,1) = " " THEN L
     F$ =  LEFT$ (WK$,S): PRINT "
     CORRECT SPELLING IS...";:X$ =
     "":UC = 0: GOSUB 480
```

For example, if you found "She was good-nachureΛ.," your "CORRECT SPELLING IS..." should be "good-natured" (without the quotes), not just "natured." If you had "she said, 'WiliaΛ will do it.' and then she...," you would need to correct the name and include the quote mark. CORRECT SPELLING IS...*enter 'William.*

I hope that this program will be useful to you as well as teach you some new programming techniques. A Spelling Checker without SPELLING CORRECTOR is like a Mo-ped without a motor. You can still get there but not as fast or as easily. Using my text processor's search routine to find marked words took 17 minutes to do the same text that SPELLING CORRECTOR did in 47 seconds.

This program can help you learn how to write directly onto the screen. How to pass numbers back and forth between BASIC and machine language. And the output filter is a very powerful tool to process any output data (not just to the screen). It works just the same to the printer or any other peripheral device. There is also an input filter very similar to it, but then that's another story.

```
360 : IF  LEN (WK$) = Y THEN RT$ =
    "": FOR S = Y TO 1 STEP  - 1
    : IF  MID$ (WK$,S,1) = " " THEN
    LF$ =  LEFT$ (WK$,S): PRINT
    : PRINT "CORRECT SPELLING IS
    ...";:X$ = "":UC = 0: GOSUB
    480
380 IF  LEN (WK$) = 1 THEN  PRINT
    "CORRECT SPELLING IS...";X$
     = "":UC = 0: GOSUB 480:L$(X
    ) = X$: GOTO 200
400 RE$ = X$
420 IF  MID$ (WK$,S,1) = " " THEN
    WK$ = LF$ + RE$ + RT$:Y = S:
    L$(X) = WK$: GOTO 200
440 IF S = 1 THEN  PRINT "CORREC
    T SPELLING? ";:X$ = "":UC =
    0: GOSUB 480:RE$ = X$:L$(X) =
    RE$ + RT$: GOTO 200
460 IF S > 1 THEN  NEXT S
480 UC = 0
500 GET K$:K =  ASC (K$)
520 IF K = 27 THEN UC = 1: GOTO
    500
540 IF K = 19 AND UF = 0 THEN UF
     = 1: GOTO 480
560 IF K = 19 AND UF = 1 THEN UF
     = 0: GOTO 480
580 IF UF = 1 THEN UC = 1
600 IF LC = 0 AND UC = 1 AND K >
    64 AND K < 91 THEN  INVERSE

620 IF K = 24 OR K = 8 THEN  PRINT
     CHR$ (92):X$ = "": PRINT "T
    RY AGAIN...";: GOTO 480
640 IF K = 34 THEN K = 39:K$ = "
    '"
660 IF LC = 0 THEN  PRINT K$;: NORMAL

680 IF UC = 0 THEN K$ =  CHR$ (K
     + 32 * (K > 64 AND K < 91))

700 IF LC = 1 THEN  PRINT K$;
720 IF K = 13 THEN UC = 0: CALL
    768: RETURN
740 IF K < 32 THEN  FLASH : PRINT
     CHR$ (K + 64);: NORMAL
760 X$ = X$ + K$:UC = 0: GOTO 500
    : REM


780 X = 0: REM **READ TEXT FILE**

800 PRINT : PRINT : PRINT "READI
    NG DISK FILE  '";NM$;"'"
820 PRINT D$;"OPEN"NM$: PRINT D$
    ;"READ"NM$
840 INPUT I: INPUT SL: INPUT TB:
     INPUT CL: INPUT CI
860 FOR X = 1 TO I
880 INPUT L$(X)
900 NEXT
920 PRINT D$;"CLOSE"
940 RETURN
960 PRINT "DO YOU WISH TO SAVE T
    HE CORRECTED FILE BACK TO DI
    SK...Y/N  ": GET AN$: IF AN
    $ <  > "Y" THEN 1140
980 PRINT : PRINT "DONE": END
1000  TEXT : HOME : VTAB 12: HTAB
    10: FLASH : PRINT "SAVING DA
    TA TO DISK": NORMAL
1020  PRINT D$;"OPEN"NM$: PRINT D
    $;"DELETE"NM$: PRINT D$;"OPE
    N"NM$: PRINT D$;"WRITE"NM$
1040  PRINT I: PRINT SL: PRINT TB
    : PRINT CL: PRINT CI
1060  FOR X = 1 TO I
1080  PRINT  CHR$ (34);L$(X)
1100  NEXT X
1120  PRINT D$;"CLOSE"
1140  PRINT : PRINT "ALL DONE"
```


"My Legend memory expansion cards enhance my Apple and are a superb product for the Apple Computer.*

I think the cards are mapped appropriately for extra storage.

The software supplied makes the cards useful even for those who can't modify DOS on their own.

I look forward to upcoming products from Legend."

Steve Wozniak
Co-Founder,
Apple Computer, Inc.

If Steve Wozniak, Co-Founder of Apple Computer, Inc., thinks this highly of Legend products, we're doing something right.

Go to your nearest computer dealer today and find out for yourself about Legend products.

The Original Hi-Density RAM Products for APPLE II

Legend Industries Ltd.

U.S.A.
Legend Industries, Ltd.
2220 Scott Lk. Rd.
Pontiac, MI 48054
(313) 674-0953

WEST GERMANY
Porter and Co.
Berger Strasse 140
6000 Frankfurt/Main 60
West Germany
49 611 439818

EUROPE
B.I.P.
22 Joseph Disjon
75018 Paris, France
(1) 255-4463.

*Apple II is a product of Apple Computer, Inc.

Circle 21 on Reader Service Card

SPELLING CORRECTOR

```
:ASM
                    ********************************
                    *                              *
                    *          CORRECTOR           *
                    *       Wesley R Felty         *
                    *                              *
                    * Call -A.P.P.L.E. : May, 1983 *
                    *                              *
                    ********************************
                    *
                            ORG   $300
                    *
0300: A9 0B                 LDA   #<SCRPR     ;Low byte of start of program
0302: 85 36                 STA   $36
0304: A9 03                 LDA   #>SCRPR     ;High byte of start of program
0306: 85 37                 STA   $37         connect DOS hooks
0308: 4C EA 03              JMP   $3EA        ;inform DOS of change of hooks
030B: CD 36 03      SCRPR   CMP   ERRTOK      ;actual program starts here
030E: D0 13                 BNE   NOT         ;GOTO "NOT" if no corrector symbol
0310: AD 34 03              LDA   TSTPOS      ;an error has been found, so load
0313: 8D 35 03              STA   ERRPOS      ;position in line where found and
0316: AD 36 03              LDA   ERRTOK      ;store it in "ERRPOS" (error position
0319: E9 80                 SBC   #$80        ;Now we make the error symbol flash
031B: 09 40                 ORA   #%01000000
031D: 20 38 03              JSR   ALPHA       ;on the screen and ? UC/LC char's
0320: 4C 3A FF              JMP   $FF3A       ;ring the bell to show error found
0323: C9 8D         NOT     CMP   #$8D        ;Char wasn't error... is it carriage
0325: D0 07                 BNE   CONT        ;return? If not, go to "CONT"inue
0327: A9 00                 LDA   #$00        ;If carriage return, initialize
0329: 8D 34 03              STA   TSTPOS      ;position being tested back to 0
032C: A9 8D                 LDA   #$8D        ;Now actually issue the return
032E: EE 34 03      CONT    INC   TSTPOS      ;Add 1 to position number & go on
0331: 4C 38 03              JMP   ALPHA       ;Print to screen the character
0334: 00            TSTPOS  DFB   #$00        ;Store here the line position test
0335: 00            ERRPOS  DFB   #$00        ;Store here position or last error
0336: DE            ERRTOK  DFB   #$DE        ;Store here token for error symbol
0337: 00            LCFLAG  DFB   $00         ;Store here 00 if no LC chip or
                    *                        ;or store 01 if LC chip present
0338: 48            ALPHA   PHA   PRINT       ;out char's to screen
0339: AD 37 03              LDA   LCFLAG      ;Check for a lower case chip
033C: C9 01                 CMP   #$01        ;01 means lower case chip present
033E: 90 04                 BCC   LC0         ;If no LC chip, GOTO LC0
0340: 68                    PLA
0341: 4C F0 FD              JMP   $FDF0       ;If LC chip print directly out
0344: 68            LC0     PLA   SAVE        ;The character
0345: C9 E0                 CMP   #$E0        ;Is character upper or lower case
0347: 90 06                 BCC   UC          ;branch if upper case
0349: 38                    SEC   PRINT       ;Lower case in inverse
034A: E9 20                 SBC   #$20        ;Convert lower case to upper case F
034C: 4C F0 FD              JMP   $FDF0       ;Display to screen
034F: C9 C0         UC      CMP   #$C0        ;if not number, then make inverse F
0351: B0 03                 BCS   INV         ;Letters but not for numbers
0353: 4C F0 FD              JMP   $FDF0       ;finally! Print out to screen
0356: 29 3F         INV     AND   #$3F        ;This is a capital so make inverse
0358: 4C F0 FD              JMP   $FDF0       ;Then print it

--End assembly--

91 bytes

Errors: 0
```

New Book

IT'S JUST GREAT

№ 1 IN THE SERIES

IJG

THE CUSTOM APPLE & Other Mysteries.

CREATE A WHOLE NEW WORLD OF POSSIBILITIES FOR YOU AND YOUR APPLE.

Dateline: California, January 1983.

J.R. Programmer, bit diddler and hardware hacker extraordinaire', wished that his APPLE computer had 6522 I/O capabilities available.

To have that one enhancement would open new vistas in data acquisition and display, complete his project, and make him millions.

"Gosh, wouldn't it be nifty?" exclaimed J.R.

But alas, the lack of appropriate hardware prevented him from accomplishing his goal.

J.R. looked everywhere in his local computer store for an interface or expansion board to solve his dilemma, but none were designed for the magnificent project he had in mind.

"Shucks . . ." he stormed, "what am I to do?"

Poor J.R.

But then, just as he was about to give up all hope, a splash of color caught his eye over in the book section . . .

The Custom APPLE & Other Mysteries.

"Oh joy, Oh joy!" cried our hero. "This book gives me the information, specifications and references I need to do the hardware enhancements, whether I am a beginner or an expert, and provides me with the basic information required to write the software, common to many of the projects, myself. This book is just crammed with all kinds of neat stuff!"

J.R. left the store, also an IJG book dealer, a very happy man.

A Hardware and Software Modification Guide.

The Custom APPLE & Other Mysteries provided J.R. with a number of data acquisition and control projects with camera ready printed circuit layouts like the 6522 application interface board, an 8-Bit D/A and A/D converter, a sound and noise generator board, an EPROM burner board, an APPLE Slot Repeater, and included information on the APPLE as a square wave generator, the control of two stepper motors, connecting two 6502 systems, and *lots lots* more.

Plug-In To Power And Get Turned On!

The Custom APPLE & Other Mysteries is available for $24.95 at computer stores, B. Dalton Booksellers and independent book dealers around the world. If your dealer is out of stock, order direct from IJG.

Include $4.00 for shipping and handling. Foreign residents add $11.00 plus purchase price. U.S. funds only please.

IJG, Inc. 1953 West 11th Street
Upland, California 91786
Phone: 714/946-5805

IJG

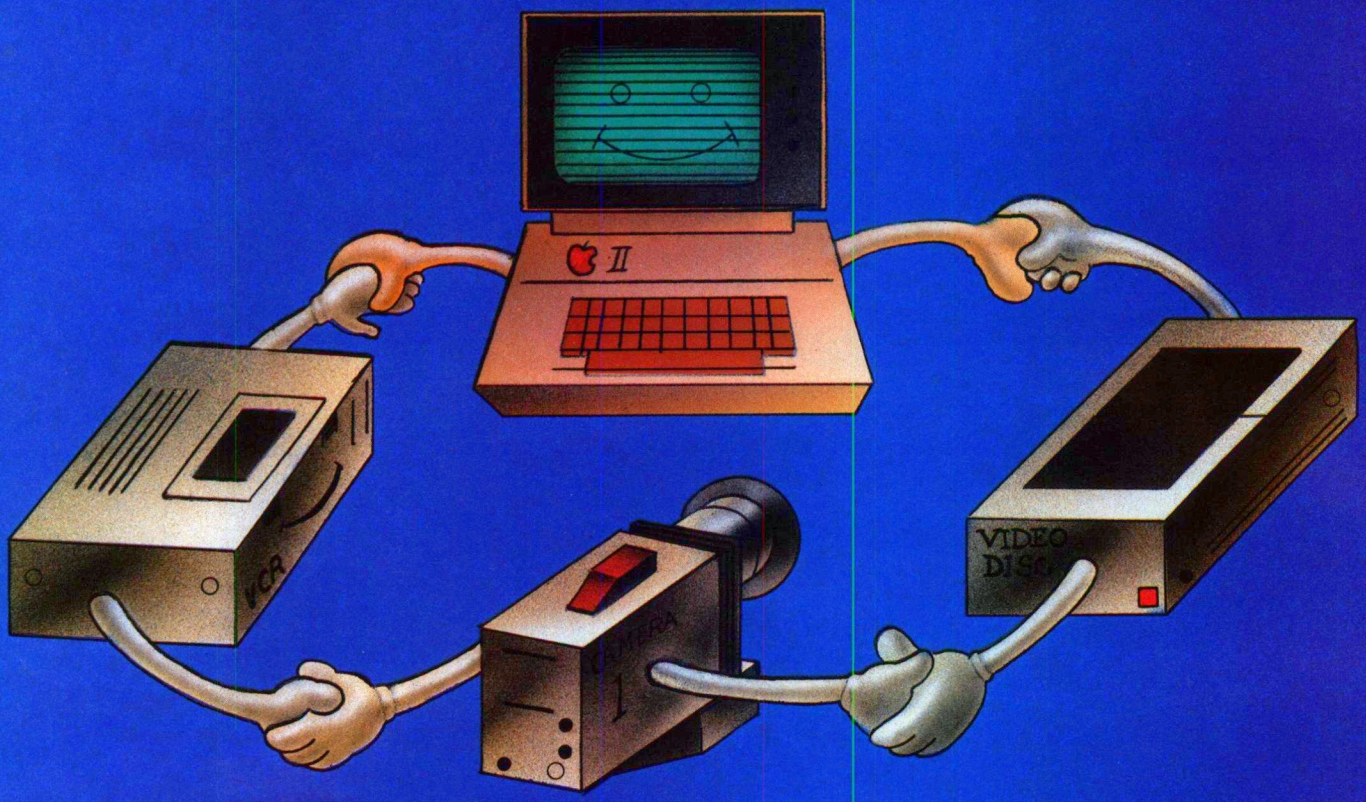
Helping You Help Yourself.

© IJG, Inc. 1982

Circle 51 on Reader Service Card

TM APPLE and APPLE II Trademarks of APPLE Computer Inc.

Now They Can Finally Afford To Be Married



And What Beautiful Offspring!

WEATHER MAPS

TRAINING

COMPUTER GRAPHICS

This marriage made possible by VAL's MicroKey System.

The MicroKey System allows the Apple II™ Computer to generate standard N.T.S.C. video, to be genlocked, and to key over an incoming video source.

VIDEO ASSOCIATES LABS
2304 Hancock Drive, Suite One
Austin, Texas 78756
(512) 459-5684

MICROKEY SYSTEM

MICROKEY IS A REGISTERED TRADEMARK OF VIDEO ASSOCIATES LABS

PATENT PENDING

™Registered Trade Mark Apple Computer, Inc.

Circle 39 on Reader Service Card

A **Call -A.P.P.L.E.** PRODUCT REVIEW:

VB-3 MICROKEYER

James L. Hockenhull/reviewer

VB-3 Microkeyer *$2400*
Video Associates Labs, Inc.
2304 Hancock Drive, #1F
Austin, TX 78756

Introduction

AN Apple][or][+, as it comes off the shelf, will not interface satisfactorily with most other video equipment. The signal emitted at the Video Out jack differs significantly from that required by electronic devices conforming to the standards set down by the National Television System Committee (NTSC), meaning virtually all professional recording and broadcasting equipment in the United States. I put forth a semi-technical explanation of the problem in my article "Video Interfacing" (*Call -A.P.P.L.E.*, June, 1982). Response to the article clearly shows that many Apple users want to take advantage of their computers' text and graphics capability as a helpmate for a wide variety of video projects. With pleasure I offer here a review of a peripheral device, the VB-3 Microkeyer, designed to convert the Apple's signal into one conforming to NTSC standards. This goody sells for $2,400. It is manufactured by Video Associates Labs in Austin, Texas, a firm which produces a line of somewhat esoteric video devices. (Note: users tend to call the company "VAL," an acronym inconveniently close to the given name of our estimable editor. For clarity's sake, I will use the initials V.A.L. here and trust that none will think I'm referring to Mr. Golding.)

What Is It?

The VB-3 Microkeyer is a complex assemblage of electronics which fits in, on, and around the Apple. Its primary purpose is to correct (or, more accurately, to rebuild) the computer's video signal to conform to the NTSC standard, a purpose it accomplishes very well. In addition, it performs some secondary functions which do much to increase its usefulness as a video tool. For example, it allows the Apple's video output to be superimposed ("keyed") over another signal which might come from a video tape recorder (VTR), a camera, a studio switcher, a video disk player, or from an on-board device which generates colored backgrounds ("mattes").

Three distinct hardware units make up the Microkeyer: the "VB-3 Circuit Board," about 5.5" X 10.5", sits on top of the Apple's power supply; the "VB-4 Slot Card," 10" long, resides in any peripheral slot (except Slot #0); and an outboard power supply, housed in a utility case 2.75" X 4"X 6." Umpty-jillion cables (well, seven actually) connect these components with each other and with the motherboard of the computer. In spite of its complexity the VB-3 is rather easy to install (see below).

Supporting software is provided on a 5¼" floppy disk, requiring DOS 3.3 and Applesoft for use. Principal among the routines is a machine language program named OBJECT which initializes the system and sets up the pointers necessary for ten ampersand (&) commands for control of the various functions of the VB-3. (As I do not normally program in Applesoft, this was my first experience with the ampersand. I'm impressed with the power it adds to BASIC.) OBJECT hides itself within DOS where it is protected from virtually all BASIC code. A general demonstration program and one which allows color adjustments to be made by means of the game paddles round out the disk's catalog. All programs exist in "regular" and "PLE" versions, the latter relocated for use with the popular Program Line Editor.

It is important to note that at the system level the VB-3 is controlled by writing to addresses ("software switches") within the peripheral I/O space. While the support software will make applications of the unit easier under some circumstances, it is *not* essential. Any programming language allowing byte-level access to the computer, as Apple's BASICs do with their PEEK and POKE commands, will be able to drive the Microkeyer, at the cost of some programming effort.

A plastic-spiral bound manual, about fifty pages long, gives instructions for installation, adjustment, and use of the VB-3. It is unpretentiously reproduced, one side per sheet, from dot-matrix printouts. Pages are not numbered and there is no index.

The disk is not copy-protected — loud huzzahs for Video Associates Labs! — and the user is told that it "...should be copied immediately and as often as necessary to assure that a good back-up disk is always available."

Installation

Setting up the Microkeyer is not exactly trivial, but should pose no problems for anyone with a bit of experience in solid-state electronics. I'm sure a rank novice could manage it, but would be well-advised to practice on something less valuable first. Nine integrated circuit (IC) chips must be removed from the Apple motherboard and replaced with plugs attached to the various VB-3 cables. An inexpensive IC puller and inserter, such as Radio Shack's 276-1574, is a wise purchase to help prevent damage to the chips.

Once the cable plugs are installed the VB-3 Circuit Board is fastened to the top of the Apple power supply with double-sided tape (a simple though not particularly elegant method), the VB-4 Slot Card is installed in any valid peripheral slot (software is set up for Slot #7, but can be modified easily) and the whole works interconnected by the appropriate cables. The satellite power supply sits somewhere on the work table. Communication to the outside world is handled by two shielded cables terminating in male BNC connectors, supplying the NTSC output signal and receiving input from an external video source, if any.

Installation instructions are very clear. The entire job should not take much more than twenty minutes. All the usual precautions regarding the handling of integrated circuitry should be followed.

VB-3 MICROKEYER

Operation

When the computer is booted on the V.A.L. disk, or the program "VB–3" is RUN from Applesoft, OBJECT is BLOADed into memory, setting up the important ampersand commands. A menu then offers three choices:

1) RUN DEMO PROGRAM.
2) RUN PADDLE PROGRAM.
3) DO YOUR OWN THING.

The smart choice, I'm afraid to say, is 3) which simply exits to Applesoft, leaving the user free to read the manual and fool around. The other two programs are all too typical of much demonstration software: obscure, confusing, and poorly thought out. (Have *you* ever tried to write a comprehensive, idiot–proof program of any kind? It is tedious work, to say the least. We should not be too harsh on those who fail to pull it off. On the other hand, when we are asked to pay a "professional" price for a product, as we are here, we have the right to expect the highest quality throughout.)

OBJECT creates ten ampersand commands. An explanation of them is an explanation of the many functions of the VB–3 device, so I will go into them in some detail.

VB-3 Ampersand Commands:

I. &SWITCH A, B

Argument A is an integer which selects among ten software switches. Argument B determines the NORMAL (1) or ACTIVATED (2) setting. The switches are:

1) INVERSE: when activated the background is displayed as white with characters in black.
3) HALF SATURATION: sets the Apple graphics to 50% of their normal intensity.
4) TEXT WIPE: displays a colored matte only behind the four text lines of a mixed text/graphics display.
5) KEY ON–OFF: turns the computer output on or off. When off, only the external video source is displayed. (Note: the Apple's output remains available in black and white at the Video Out jack.)
6) B & W MODE: disables Apple's color.
7) GENLOCK: synchronizes the output signal to that from an external video source.
8) PROC AMP (PROCessing AMPlifier): replaces the synchronizing information on the external signal.
10) KEY FILL: replaces *all* Apple colors with the color of the background matte.
11) BACKGROUND MATTE: chooses between a colored or black background. The colors from the on–board matte generator fill the entire video screen rather than just the undersized area of normal Apple display.
12) BASE VIDEO: selects the background from either an external video input or the matte generator. If the external source is chosen, Apple graphics can be keyed over normal TV pictures.

Switches #2 and #9 are not described. #2 appears to be waiting for forthcoming software which will, by combining graphics pages 1 and 2, allow the Apple to generate graphics on the full screen with a resolution of 360 X 240, an exciting prospect.

II. &ADJCLR A, B

This command controls several factors which determine the colors of the display. Argument A will be the number of the desired adjustment, argument B specifies the setting. The adjustments are:

15) APPLE HUE: the colors of the Apple graphics. Argument B can range from 0 to 359 (degrees). The effect of this command is much like fiddling with the tint control of a TV set or monitor.
16) MATTE HUE: the colors of the full–screen background matte. Argument B again ranges from 0 to 359. The matte color will appear wherever the Apple would normally display black.
17) MATTE LUMINANCE: the amount of white mixed with the matte hue.
18) MATTE SATURATION: the intensity of the matte hue. In these two latter cases, argument B can take on values between 0 and 100 (percent).

As the arguments are single bytes, they can really only take on 256 different values. Still, this means that by combining the possible settings of hue, luminance, and saturation, some 2,560,000 matte colors are available! Of course the human eye cannot distinguish between all of these. Here, switches 13 and 14 are not described.

III. &INITIAL: resets all switches and controls to their "normal" settings.

IV. &SLOT A: selects a peripheral slot for the VB–4 Card. As it comes, the software expects the card to be in Slot #7.

The following four ampersand commands manipulate "presets," twelve areas of memory which can store the settings of all VB–3 controls.

V. &FILLSET A: save all settings into preset A where A is an integer between 1 and 12.

VI. &PRESET A: set all controls to the values stored in preset A.

VII. &KEEPPRESETS and

VIII. &FETCHPRESETS: save or load twelve presets from disk.

IX. &PADDLE: this useful command allows the four color adjustments to be made by Game Paddle #0.

X. &CLRINIT: changes the initialization settings in the OBJECT program.

I repeat that although these ampersand commands will make life a lot easier for the Applesoft programmer they are not strictly necessary for operation of the Microkeyer.

User Comments

To get right to the crux of the matter, the VB-3 Microkeyer works. It puts out a fully interlaced signal that looks mighty close to NTSC standard on a "scope." Professional television equipment accepts the output with no problem. Here, at last, is a way to get Apple text and graphics onto videotape or onto the airways. Definitely good news for many of us!

A dry listing of its many functions does not give an adequate picture of how the device enhances Apple's output. The matte generator fills the screen with color, a big improvement in itself. By adjusting matte color and Apple hue, colored text can be displayed on colored grounds. High-Res graphics take on a wider range of color combinations. And all this can be done "on the fly" from a program. In other words, the VB-3 does more than just transform Apple's video signal: it provides a tool for graphic creativity.

However, anyone used to the rock-solid, high-resolution image of a computer display on a good monitor will notice some vertical twitching or "jitter" in the VB-3 picture. It ranges from barely noticeable to aggravating, depending upon the colors and shapes being shown. Arthur Bell of V.A.L. assures me that this is an inevitable concomitant of the NTSC system. I've been keeping my eye on the output of broadcast quality character generators, the devices that do most titling for modern television, and have noticed jitter in their displays, too. The scale of an image is critical. Eighty-column characters, for example, are simply beyond the resolution of the television system and will be illegible.

Documentation is a constant problem in all parts of the computer industry. (Have you ever tried to read the UNIX manual?) It is difficult for the designers of a product to fully appreciate the ignorance of those dealing with it for the first time. The Microkeyer manual runs the gamut: installation instructions are very clear indeed, as is the other technical information, but the rest seems to be a bit muddled. Program listings are provided for the mediocre demonstration programs but not for the all-important OBJECT code. And early editions, which are in no way distinguished from later versions, contain some aggravating address errors in the section on direct control of the device. The manual does not yet do justice to the product.

The personnel of Video Associates Labs have proven to be very friendly and helpful, rightly proud of their product and concerned about its success. A purchaser of the VB-3 would be assured of company support.

Of course the question must arise: Who is going to buy this thing, this peripheral that costs more than the computer itself? Its obvious targets are small television stations and similar video production operations. In this market it is certainly competitive, considering that a good character generator runs in the neighborhood of $15,000. (Nice neighborhood!) The Microkeyer should be a godsend to such outfits, as it opens the intriguing possiblity of easy, low-cost animated graphics, with a powerful general-purpose microcomputer thrown into the bargain.

Conclusion

The VB-3 Microkeyer is doubly impressive. It not only converts the Apple][+'s video signal to NTSC standard, but provides a number of important secondary functions which help to make the computer display more useful and better looking. Although electronically complex, it is easy to install and use. It can be driven from Applesoft programs using the support software, or can be controlled directly by any programming language allowing byte-level access to the Apple system. All manner of small video production operations should find it an invaluable tool.

Additional Product Information

Hardware:

One printed circuit board, one peripheral slot card, one outboard power supply, connecting cables.

Software:

One system diskette containing object code and Applesoft programs with variants for use with PLE. (Not copy protected.)

Documentation:

Spiral-bound manual, ca. 50 pages, containing information on installation, operation, support programs, memory usage, alignment and adjustment. No page numbers, no index.

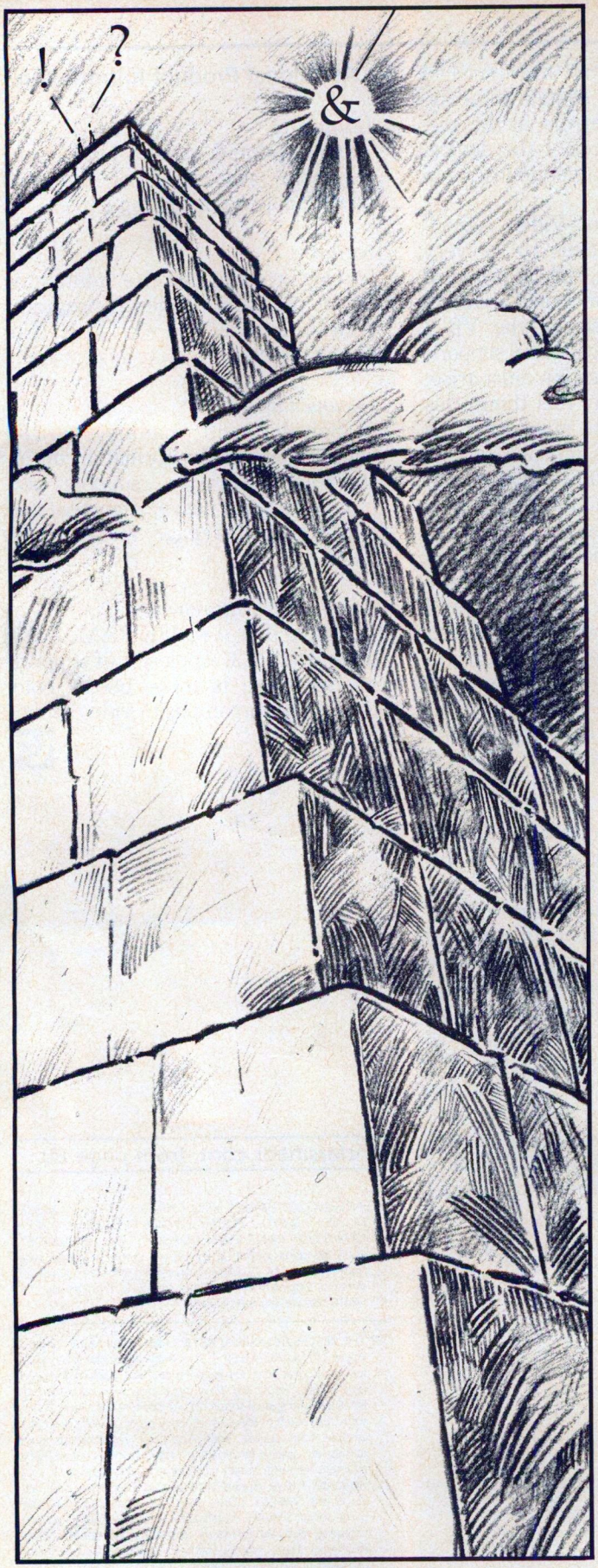
Equipment Requirements:

Apple][or][+, 48K RAM, 1 disk drive, DOS 3.3, Applesoft (if supporting software is to be used.), color monitor with composite video input.

Unclassifieds cont. from page 80

ENGINEERING SOFTWARES offers APPLESOFT PLUS for MORE MONEY - SPEED - PROTECT YOUR PROGRAMS TOO - CAN'T LIST, PRINT EVEN DUPLICATE IT! - APPLESOFT PLUS lets users with 1.0K memory and execute larger programs. All for $44.95. Other Engineering softwares available. Write to Engineering Softwares, Box 872, Duncanville, TX 75116. Dealer inquiries invited.

GET THE BEST AUTO-REPEAT KEYBOARD ENHANCER for your Apple][/][+ AUTOREP 1.3 — Fully adjustable time delay for auto-repeat after any keypress plus adjustable high-speed repeat with "Rept" key. Easy installation with no soldering. Small size and special shape do not interfere with long accessory cards (such as 80-column). Low-profile on/off switch accessible from outside your Apple for convenience with games, etc. Special terminal for easy connection of shift-key adapters. Highest quality components throughout, 1 year warranty. Fits all Apples with plug-in encoder board (late Rev-6 on). SHIFKEY 1.0 — The neatest way to make the game socket shift-key modification. Installs securely with no soldering. Actually increases access to the game socket. AutoRep 1.3 only $19.95. ShiftKey 1.0 only $4.25 (specify for use with or without AutoRep). Prices include First Class Air Mail postage throughout U.S. for fastest delivery send cashier's check, m.o., or VISA/MC number (with exp. date). Personal checks subject to slight delay. APPLE CLUBS please write for discounts. Dealer inquiries invited. PACIFIC COMPUTER ACCESSORIES, Dept. UCA, P.O. Box 25730, Honolulu, HI 96825.



We've moved ahead a byte in time since the Tower of Babel. And at least three things have improved.

Here they are:

Thing Number One: Our Communication's Improved.

With AMPER-MAGIC™ there's no need to know machine language to benefit from its power.

To your Applesoft™ programs you can attach slick, finished machine language routines (referenced by name, not by address, no less) in seconds. No separate BLOADing. And then, with AMPER-MAGIC commands, you can pass variables back and forth directly, just as you would with built-in Applesoft commands.

So, name your routine, perform the append procedure (only once please), and AMPER-MAGIC takes over — generating machine language speed where it counts: in *your* programs. Even your commercial programs. And, if for some reason you want to remove the routine, AMPER-MAGIC can do this too.

Thing Number Two: Our Communication's Improved.

AMPER-MAGIC also gives you access to hundreds of relocatable machine language routines, including those from magazines and other libraries, and - most notably - those from AMPER-MAGIC itself, which has over fifty routines of its own in two volumes.

Some of AMPER-MAGIC's routines are:

Find substring	Swap variables	GOSUB, GOTO a variable
Speed up Applesoft	Input anything	Search string array

And, of course, there are others too, for business, educational, recreational and word-process uses.

With AMPER-MAGIC you attach and access. It's really that simple. You can attach an unlimited number of commands to any program and you can move any set of commands as a unit to any other program.

A marvelous simplicity, wouldn't you say?

Thing Number Three: Our Communication's Improved.

Compatible with AMPER-MAGIC you can get the AMPER-MAGIC COMMAND LIBRARY. Already available is Volume 2, which contains twenty-seven machine language routines devoted to information displays and output capabilities.

With Volume 2 you get: a WAIT command, which specifies time in ordinary tenths of seconds either as a variable or as an expression which can be controlled by the program; the most powerful PRINT USING command yet devised; and, of course, twenty-five other commands. Future volumes include INPUT USING, fancy memory management and various sorting routines.

Imagine!

Only $75.00 for the original AMPER-MAGIC by Bob Nacon

Only $35.00 for AMPER-MAGIC COMMAND LIBRARY, Volume 2

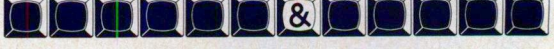
Plus shipping. MC and Visa accepted.

For those of you who would like AMPER-Magic's routines in your own programs, just ask for Anthro-Digital's no cost licensing policy.

AMPER-MAGIC and AMPER-MAGIC COMMAND LIBRARY are trademarks of Anthro-Digital, Inc.

Applesoft is a trademark of Apple Computer, Inc.

Amper-Magic™



See your dealer or contact us.

Anthro-Digital. 103 Bartlett Avenue, Pittsfield, MA 01201

Telephone (413) 448-8278, Telex 467622

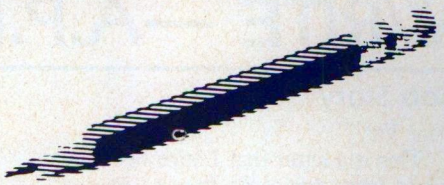
"The Company That Interfaces People and Computers"

Also available from Anthro-Digital: Versacalc (several versions), Visicalc Formatting Aids, The Executive Secretary, The Personal Secretary, and Executive Speller, Hebrew II and Hebrew II Plus, S-C Macro Assembler, QuickTrace, Amper-Magic, The Rental Manager, F.A.R.M. accounting packages, The Performance Manager, Omniscan VideoDisc interface, and Flipper.

DEALERS INVITED

Circle 42 on Reader Service Card

WRITE -A.P.P.L.E.



the readers

When is an LS not an LS?

Dear Editor:

HERE is some information that may be of great interest to some of your readers using or repairing the Game I/O port on their Apples. It concerns some errors in the documentation about this port.

The *Apple II Reference Manual* (p. 100) describes the annunciator outputs of the Game I/O port as "standard 74LS series TTL outputs" and the schematic diagram shown (p. 114) shows the outputs to be those of a 74LS259 8-bit addressable latch. However, the *Applesoft II Basic Programming Reference Manual* (p. 135 refers to these outputs as "open-collector.' The 74LS259 is not an open-collector device.

Neither of these quoted assertions is correct. When a hardware problem developed with an Apple II on which I was working, I searched for the 74LS259 described in the schematic. It was not to be found. The actual output device eventually found in the Apple (and in all 10 other Apples I checked) was a 9334. This chip, equivalent to a 74259, is not a 74LS series device, nor does it have open-collector outputs. It is a 7400 series device. It has twice the output current capability of the 74LS259, and it has regular Darlington totem-pole outputs.

These are not the only errors in the documentation concerning these outputs. All of the statements listed in the reference manual for setting these outputs (p. 135) are in error. The statements listed as setting the outputs to TTL high, set them low, and vice versa. Furthermore the maximum output low current listed, 1.6ma, (p. 135) is not the correct one for either of the devices mentioned above. The actual rating for the 74259 is 16ma, and the rating for the 74LS259 is 8ma. The value listed is not the output low current, but the input low current for the 74259.

These discrepancies in the Apple documentation could cause quite a problem for someone wishing to interface with the Game I/O port.

Gary J. Sullivan
Graduate Teaching Assistant
Electrical Engineering Dept.
Speed Scientific School
University of Louisville
Louisville, KY 40292

Control D Returns

Dear *Call -A.P.P.L.E.*

A brief note on [Ctrl-D]. Staff writer Ralph Swerdlow's "Controlling Control D," like many, many others, suggests using D$ = CHR$(13) + CHR$(4) to define Control-D. The reasons for this, of course, is to assure that CHR$(4) winds up in column 1 (forced there by the return, which is CHR$(13)) — since column 1 is the only place it will work.

The definition of Control-D will really louse up text files in certain circumstances. Specifically, if you have a file open and write-enabled, and then you issue a read command to a second file for input, the CHR$(13) return character will be written into your write-enable file. These extra returns in your text file will lead to the collapse of most applications.

Dave Garson
North Carolina State University
P.O. Box 5305
Raleigh, N.C. 27650

CURSES, FOILED AGAIN! Seriously, that is one aspect that never occurred to us. Readers would be well advised to keep this in mind.

Epson CP/M Patch

Dear Editor:

I have recently interfaced a Daisywriter to my Apple using an EPSON MX series parallel interface kit type 2. I had some problems using it with CP/M. There is a patch in the manual, page 28, for use with the Z-80 card, through the use of DDT.

However, this patch is not permanent and has to be done every time you boot the system. Using DU-V75, or any other CP/M disk utility, it is possible to patch CP/M permanently. There may be an easier way, but I am a novice and this is the best I have found...

In any event, if this can be of any utility to EPSON owners, the address to modify (DD2F) is found on track 2 sector 17 (for a 54K system). Using DU-V75, first read track 2, sector 17, then CHange address Hex 2F (CH2F,31) from 3E to 31 and write it back to disk (W).

If there is a better way to do it, I would appreciate the information.

Patrick M. Dubois
7720 Malahat Avenue
Richmond, B.C. V7A4H2
Canada

Another Trap

Dear Editor,

The letter from George Miller (*Call -A.P.P.L.E.*, Feb. '83, p. 66) about incompativilities between the Novation modem and some programs reminded me of a similar problem we had with the Apple at work. Dr. Miller is right in thinking that interrupt requests generated by the modem are involved, but it's not the fault of the modem. The real problem is a bug in the monitor ROM. Perhaps I can shed some light on the cause and possible solutions.

Like Dr. Miller, I tried to show off Visicalc's capabilities and found it wouldn't run. As I recall, it crashed into the monitor, displaying register contents, when I tried to save a file to disk. Later I found that Superscribe II did the same thing during boot-up. I also found, as he did, that the monitor's G command didn't work. After laborious (but educational) troubleshooting I found that the interrupt request pin of the 6502 had been bent under the chip during assembly. It floated low, giving a continuous interrupt request. Everything worked fine as long as interrupts were disabled.

At boot-up, the Apple plugs the maskable interrupt vector at $03FE with the address of a routine that displays registers and exits to the monitor. Hardware that uses interrupts must patch that vector to point to its own interrupt handler routine. If not, and interrupts are enabled, the program will crash into the monitor. That's what happened to Dr. Miller and to me.

OK, so how can interrupts be enabled? Obviously with the processor's CLI command (clear interrupt disabled), but that's not the only way. The monitor's G command loads registers including the processor status (P) register from memory locations $45-$49. It's assumed that the registers' contents have been stored here previously. The bug is that they are *not* stored here on normal entry (CALL -151), so the G command loads the registers with garbage (and you thought the G stood for Go). Interrupt disable status is controlled by the state of bit 2 of the byte loaded into the P register; if it's a zero, interrupts will be enabled. And it *is* zero. A side effect of a standard Apple DOS boot is that the P storage byte ($48) is zeroed as part of a DOS address.

Everett Ogden
16 Herber Avenue
Delmar, NY 12054

WRITE —A.P.P.L.E.

Sob Story

Dear Val,

This is a letter that I have been meaning to write for some time to express a frustration I have with copy protection that I haven't seen written about, namely having to use nine different disks to use three business software packages that I use daily. Back in 1978 when I first started using the Apple there wasn't much available in the way of business software and as time went on I was thrilled to see packages become available that really did make the Apple a useful tool in the work environment. At first it didn't mean much to me that many of these packages were copy protected as I felt the authors had a right to protect their product. Over the last few years, I have increased my expectations of what I would like to be able to do and now find it very frustrating that each of these packages uses a different DOS and can't be accessed without rebooting the system.

The system I currently use has 64K, a Videx 80 column board, and Keyboard Enhancer along with several other peripherals. The business software that I have come to depend on includes Visicalc — the electronic spreadsheet, Visiplot — the graphic package that works on Visicalc files, Zardax — an excellent wordprocessor, and Synergistics Data Reporter — a data base management program. To use these programs and make use of the 80 column board and keyboard enhancer requires no less than nine diskettes and a multitude of boots. Let me give you an example of what it takes to use these programs. . .

Visicalc:

1) To use the Keyboard Enhancers macros with this I must first BRUN a macro download program on one disk.
2) To make use of the 80 column board I must use the Videx "Visicalc Pre-boot disk".
3) Then boot on the Visicalc master disk.
4) Then insert my Visicalc data file disk.

Now to use Visiplot:

5) Insert Visiplot master disk and boot.
6) Re-insert Visicalc data disk to work on files.

To move to Data reporter:

7) Reboot on Data Reporter Master diskette.
8) Insert Data Reporter application disk.

To move to Zardax word processor:

9) Insert Zardax master disk and reboot again. (Fortunately, this one has the Videx 80 col. & Keyboard Enhancer patches included but still has a custom DOS.)
10) Insert yet another disk to store my files on.

Each of these programs is a great time saver and I can't imagine going back to the days before I had them to aid me in my work, but it seems to me that for the computer to really become an integrated system that it should be easier to use and require less disk handling!

If I had access to the code from each of these programs and the files on the disks, I would first move all of the mother programs on a single diskette and keep them in disk drive 1. Then I would build a menu that would allow me to move between the programs and exit from one to the others only changing a file disk to drive two as necessary. This would lead to much less wear on my disk drive doors and make the transit time between applications considerably shorter, as there would be no need to reload DOS each time.

As I said earlier, I think the authors are entitled to receive a fair return for their work, but I no longer feel that copy protecting their disks is accomplishing this, especially in the business software arena where the documentation is often as important as the diskette, and where bit copiers have rendered most of the copy protection schemes used by these packages useless.

The user market is becoming more sophisticated as the software vendors show us more and more ways that we can use the computer as a tool in our daily work. We've come a long way since 1978 and I call on the software vendors to work on making their software more compatible within the Apple environment so we can take advantage of all that power with a reasonable amount of convenience.

David Taylor
Pacific Science Center
200 Second Ave. N.
Seattle, WA 98109

ALL POINTS WELL TAKEN. Progress takes time, but we have seen some producers at least unprotecting their software. We find it hard to believe that theft is a major problem on business software with voluminous documentation.

A.P.P.L.E. Source

Dear Joff:

Many thanks for the prompt work you did getting me up on the *Call -A.P.P.L.E.* Source System. You've got a good thing going there, and I'm looking forward to using it frequently.

The IRQ Screen Dump Routine has proved most useful. I needed such a routine for an instruction manual I'm writing to list the screens without interrupting the program or messing up the display with scrolling or other problems commonly associated with screen dump programs. I'm trying to come up with some other applications for this new capability in my Apple][.

I can't WAX as enthusiastic about the terminal program. I found it a bit fragile, some things didn't work at first, and having to directly address the micromodem is not terribly convenient. For the last couple of years, I've been using Craig Vaughan's bits package. While it isn't perfect by any means, it does have some features that you might consider adding to Remember II. Having a telephone number file, using ESC to move between the terminal mode and the main menu, and having the overall control program in BASIC so I can modify it are strong advantages.

However, the price of your system is right, and it certainly is a useful system with a number of good features. I'm sure a number of Apple members will get good use out of it. Could I suggest that you put the notice about printing out the instructions at the front of the file or on the source screen? That way, the user doesn't have to read most of the way through on the screen before finding out he/she can read a printout instead.

I applaud your putting three months of the magazine articles on the source for immediate use by the members. Your idea of putting various public domain software up is also an excellent service to the members. Order entry should also prove to be a great convenience for users like myself, who live on the other side of the country from you.

Incidentally, I'd like to put in a plug for Screenwriter II as a word processor you might look at in taking files off the system and modifying them for use as programs, files, printouts and the like.

Your PEEK/POKE routine will, I suspect, become the standard that you suggest. Score another one for you in meeting the needs of the membership!

Harry Brawley
19 Pelham Road
Weston, MA 02193

Military Applications

Dear Editor:

The 175th Medical Brigade, California Army National Guard had formed a "clearing house" for Military Users of personal computers. The unit has established an Apple][software library of military applications as well as a data base of personnel and units who are using *any* personal computer. Anyone interested may feel free to contact Maj. Jack L. Espinai, the clearing house project officer at the address given.

Jack L. Espinai
Maj. CM CAL ARNC
Project Officer
California Army National Guard
Headquarters 175th Medical Brigade
3250 Meadowview Road
Sacramento, CA 95832

Changing Drives

Dear Val,

I often have a program in memory that accesses files on a disk and that disk is on the inactive drive. This necessitates either switching disks or running CATALOG to change the DOS pointer to the other drive. The following escape code macro for GPLE alleviates the problem. It simply toggles the byte in DOS that points to the active drive. I use ESC-N for my code value since I never renumber any integer programs. Of course, any of the other ESC.codes would work as well.

```
'P = 3 - PEEK(-21912):POKE
-21912,P:PRINT PM
```

(Note: the final "M" is a control character)

Muffin Bake

An Open Letter of Confession:

Forgive me, for I have goofed. One of the unwritten laws of the Apple-ing is to search and find the uninitiated, the unbelievers and bring them to know the light that is power, (on the lower left side of the keyboard, just below the shift key). It is this law that I have broken.

Were it mere neglect of responsibilities, or simple distraction and forgetfulness, I could have sought quiet penance in a page of manual hex to dec conversion. But no!!! My error was too great. I let one get away.

I settled down to a comfortable evening of computing, and had booted GPLE. My wife came with a miscellaneous question and during my reply she interrupted me with an inquiry regarding one of the GPLE program titles appearing on my monitor screen.

It seemed odd, as she has always had ABS(0) interest in the computer. Her skills were in the culinary arts. (Although there were other considerations in my proposal to her.) Her forte was more toward the inexact, improvisational, (kitchenal), sciences of the delightful varieties of sweet, hot, sour and spicy foods of her native Thailand. For her to have shown an interest in the computer was quite a change.

I dutifully, albeit simply, explained how the program would allow the storage of the 13 to 16 sector conversion program in peripheral memory.

"Oh," she said. And left the room.

I was nonplussed. But I rationalized that I had done my duty, and that this first spark of interest in the computer was certainly an auspicious harbinger.

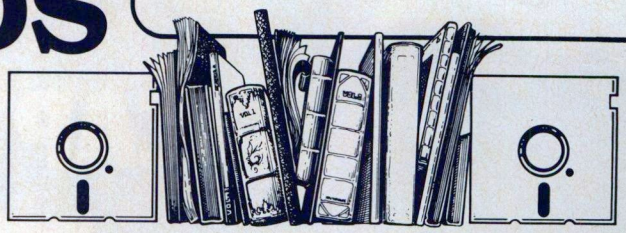
Then it hit me. I had blown it. Totally missed the boat. Not only broken, but reviled one of the unwritten laws.

The program title that my wife, the master of gastronomy, had read from the screen?

"Fix muffin".

Stan Morris
5156 W. 134th Pl.
Hawthorne, CA 90250

BOOKENDS™



The Reference Management System

Sensible Software would like to introduce you to BOOKENDS, a revolutionary new system designed to manage your references. BOOKENDS will take the guess work out of hunting for lost articles or information. Think of it as a personalized, state-of-the-art catalog system.

Pay for BOOKENDS. And not for your time.
BOOKENDS tracks down articles, magazines, and books for you quickly and effortlessly. It even prepares professional bibliographies for you. If you've ever spent time looking for important information, then BOOKENDS is for you.

BOOKENDS remembers for you.
The purpose of BOOKENDS is to keep track of information from articles and books so you don't have to. BOOKENDS works with your Apple Computer, and is menu-driven for ease of use. It has a word processor quality editor which supports upper and lower case entry and display, and also allows you to re-type just your typos, not the entire entry.

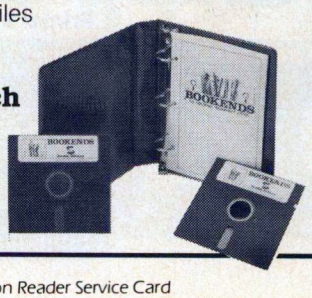
BOOKENDS allows you to store the author, title, journal, volume, page number, date, publisher, and keywords (all up to 255 characters), and an abstract (to 720 characters).

BOOKENDS also permits you to chain your reference files together, to contain any number of references you might have.

Eliminate the guesswork from your search
BOOKENDS eliminates most of the guesswork from your data search because it finds your stored information quickly and effortlessly. References can be searched for quickly by author, portions of titles, or by the keywords of your choice. And if you forget the keywords or the author, don't despair. BOOKENDS provides you with a complete, alphabetized list of the keywords and authors in the data base.

BOOKENDS is your state-of-the-art card catalog system.
BOOKENDS is particularly innovative because it can present you with professionally produced bibliographies that can be printed or used directly with your word processor. You have complete control of printouts, from simple lists including an abstract, up to professionally formatted, formal bibliographies suitable for inclusion in your word processor. When retrieving your references, the bibliography can be sorted by author, keyword, or title.

Give up the search.
Put your library in BOOKENDS. $124.95



Sensible™ Software, Inc.
6619 PERHAM DRIVE
WEST BLOOMFIELD, MI 48033
(313) 399-8877

Copyright 1983 - Sensible Software, Inc.
APPLE is a Registered Trademark of APPLE Computer Co.

Circle 31 on Reader Service Card

IMAGINE IT...

CAPTURE IT.

**COMPLETELY REDESIGNED.
NOW, THE GRAPPLER + .**

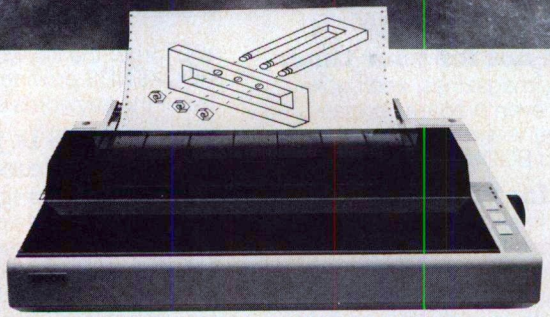
The original Grappler was the first graphics interface to give you hi-res screen dumps from your keyboard. The new Grappler + adds flexibility with on-board printer selection and 23 different commands for text and graphics. Exclusive *Dual Hi-Res Graphics* allow a side-by-side printout of graphics pages 1 and 2.

The Grappler + is compatible with the Apple II, II + , IIe and III* computers. Its extensive printer menu has been expanded to include the Apple Dot Matrix, Okidata 84 and Star Gemini, along with most popular printers. In addition, the IDS Grappler + is currently available with color capability, including color graphics screen dumps.

UP TO 64K BUFFER OPTION
An optional Bufferboard can now be added to all existing Grappler and Grappler + interfaces. See your Apple Dealer for details.

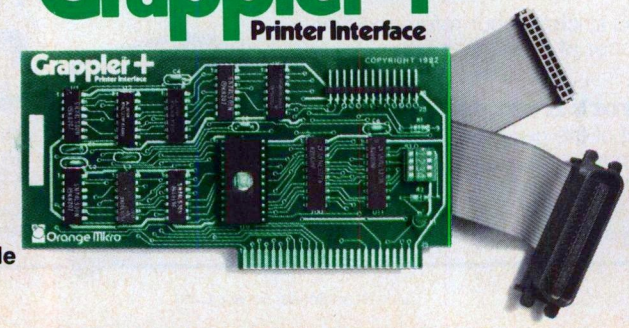
***Requires additional software driver.**

© Orange Micro, Inc. 1983.



ACTUAL APPLE II PRINTOUT USING GRAPPLER AND EPSON MX100.

With The Grappler™ + Printer Interface



Apple is a registered trademark of Apple Computer, Inc.

NOW AVAILABLE THE BUFFERED GRAPPLER +

The best of both worlds... the Buffered Grappler +

All of the popular Grappler + features with the time-saving benefits of a printer buffer.

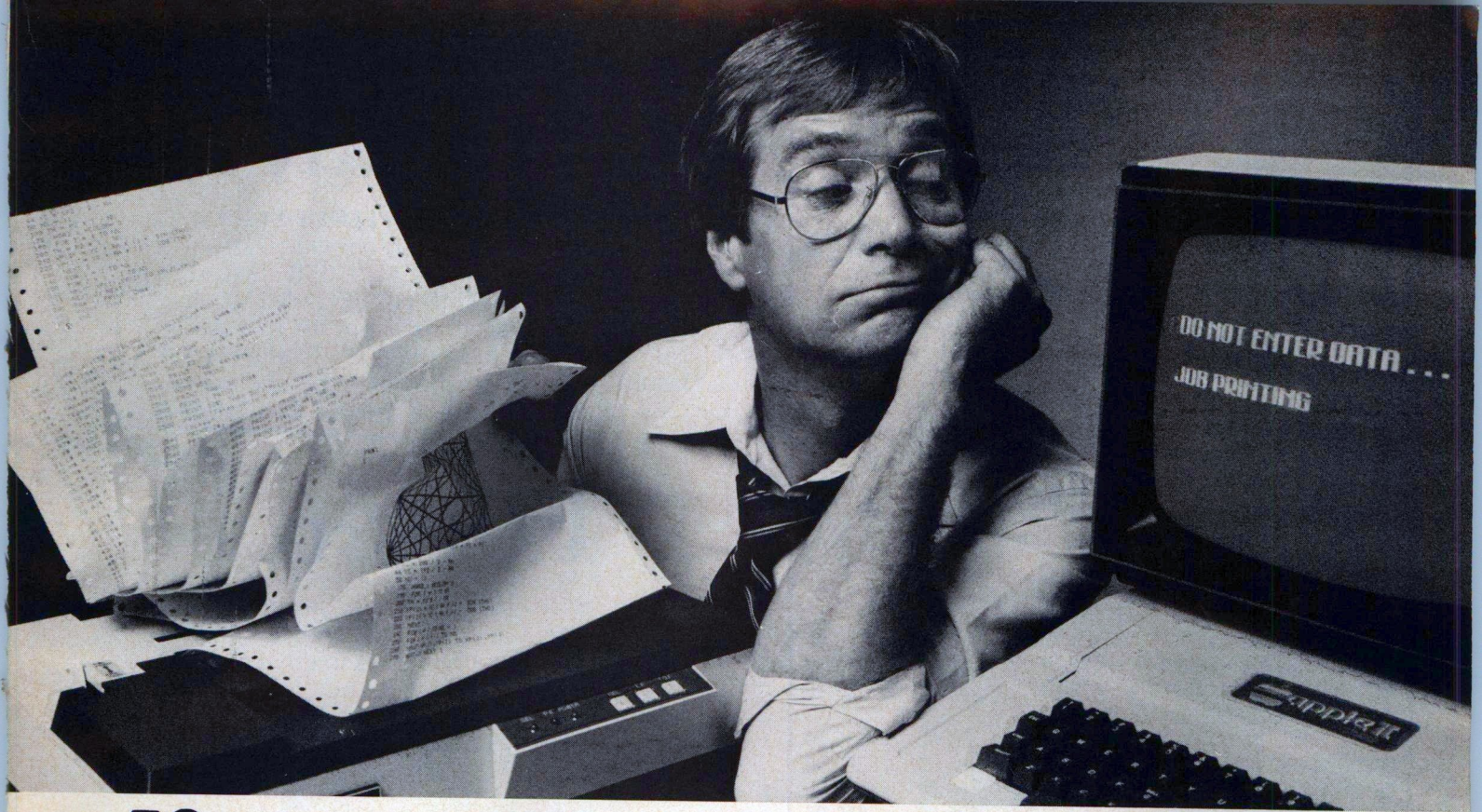
- **16K of Buffer**
- **Expandable to 64K**
- **Interfaces with all popular dot matrix printers**

Make the most of your Apple and printer, with the Grappler + or the Buffered Grappler + .

Orange Micro inc.

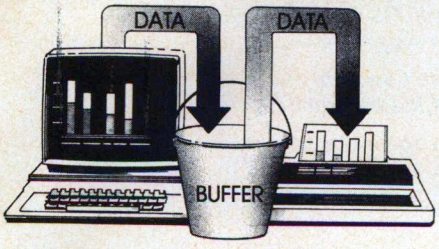
**1400 N. Lakeview Ave.,
Anaheim, CA 92807 U.S.A.
(714) 779-2772 Telex: 183511 CSMA
Foreign Dealer Inquiries Welcome**

Circle 23 on Reader Service Card



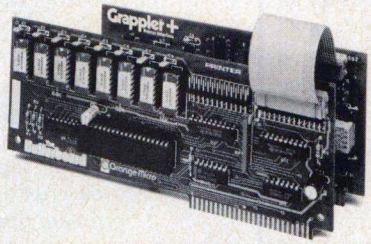
If your printer uses your Apple® more than you do, you need The Bufferboard™.

If your Apple is locked into the "PRINT" mode so much that you've taken up solitaire to kill the boredom, you need a buffer. And if your computer is the Apple II or III, the only buffer for you is The Bufferboard. Expandable to 64K of storage, The Bufferboard stores an instantaneous **bucketful** of print data from your computer. Then it feeds the data to your printer at its own printing rate. Your Apple is set free from driving your printer and is ready for more data from you.



Take your existing interface—and buffer it!

Only The Bufferboard has a simple Interface-Docking System. No bulky boxes or expensive power supplies are needed because The Bufferboard fits right into your Apple—and **docks** onto your existing printer interface. The result is convenient



and economical buffering of most popular printer interfaces, including the Grappler + ™ interface, Epson interface, and Apple printer interface. Thirty seconds and a single hook-up are all you need to end the printer waiting game forever.

Up to 20 letter-size pages stored at a time.

The Bufferboard comes standard with 16K, and is expandable to 32K or 64K of buffering capacity with the addition of memory chips. This "bucket" will hold up to 20 pages of a print job, allowing you freedom to use your Apple.

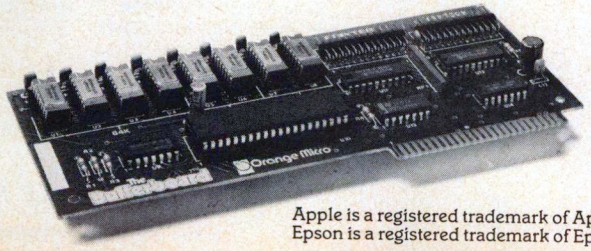
The Bufferboard—designed exclusively for the Apple Computer.

Specifications:

• Versions for Grappler + interface, Epson interface, Apple interface, and other popular printer interfaces • 16K buffer standard • Upgradeable to 32K or 64K • Automatic memory configuration • Automatic self test • Includes interface docking cable.

The Bufferboard is made by Orange Micro, Inc.; the same people who brought you the popular Grappler + printer interface. Both the Grappler + and The Bufferboard are now available at your local Apple dealer.

Orange Micro Inc.
1400 N. Lakeview, Anaheim, CA 92807
U.S.A. (714) 779-2772
TELEX: 183511 CSMA



The Bufferboard™
For Apples and Printers

Apple is a registered trademark of Apple, Inc.
Epson is a registered trademark of Epson America, Inc.

© Orange Micro, Inc. 1983.

Circle 24 on Reader Service Card

Laing Electronics

10382 STANFORD AVENUE, SUITE E-203
GARDEN GROVE, CALIFORNIA 92640
PHONE (714) 534-4216

MICROSOFT 16K CARD $69.00

★ HARDWARE FOR THE APPLE II ★

	LIST	YOUR COST
HAYES MICROMODEM II	379.00	263.00
HAYES SMARTMODEM (RS-232)	279.00	199.00
NOVATION APPLE CAT II	389.00	268.00
MICROSOFT Z-80 SOFTCARD CP/M	395.00	219.00
VIDEX VIDEOTERM CARD - 80 COL.	345.00	225.00
WIZARD 80 COLUMN CARD	249.00	149.00
WIZARD 16K RAM CARD	95.00	64.00
WIZARD BUFFERED PRINTER INTERFACE (16K BUFF + GRAPHICS, OKI, EPSON, NEC, OR ANADEX)	179.00	129.00
AMDEK VIDEO 300 GREEN MONITOR	199.00	143.00
NEC HI-RES GREEN MONITOR (JB1201)	285.00	153.00
MICRO-SCI A2 APPLE COMPATIBLE DISK DRIVE W/O CONTROLLER	395.00	235.00
MICRO-SCI A2 DRIVE WITH CONTROLLER	495.00	306.00
OKIDATA MICROLINE 83A PRINTER	899.00	665.00
NEC PC-8023A PRINTER	695.00	450.00
KENSINGTON SYSTEM SAVER FAN	89.95	69.00
GIBSON LIGHT PEN / LPS II (PENTRAK)	345.00	239.00

★ VERBATIM DISKS ★

100 DATALIFE 5¼'' 40 TRACK	450.00	253.00
50 DATALIFE 5¼'' 40 TRACK	225.00	132.00

★ SOFTWARE FOR THE APPLE II ★

PFS: FILE (SOFTWARE PUB.)	125.00	76.00
PFS: REPORT (SOFTWARE PUB.)	95.00	58.00
PFS: GRAPH (SOFTWARE PUB.)	125.00	76.00
DB MASTER (STONEWARE)	229.00	147.00
DB UTILITY PACK (STONEWARE)	99.00	64.00
DB UTILITY PACK II (STONEWARE)	99.00	64.00
THE HOME ACCOUNT (CONTINENTAL)	74.95	47.00
VISICALC 3.3 (VISI CORP)	250.00	179.00
MULTIPLAN (MICROSOFT CP/M)	275.00	167.00
MULTIPLAN (MICROSOFT DOS3.3)	275.00	167.00
WORD STAR (MICRO PRO CP/M)	495.00	299.00
SCREEN WRITER II (ON-LINE)	129.95	79.00
WORD HANDLER (SILICON VALLEY)	199.00	124.00
FORTRAN-80 (MICROSOFT CP/M)	195.00	125.00
ASCII EXPRESS PRO. (SOUTHWESTERN DATA)	129.95	85.00
Z-TERM PRO (SOUTHWESTERN DATA)	149.95	98.00
TRANSEND I (SSM)	89.00	61.00
DATA CASPTURE 4.0 (REG. VER.)	64.95	42.00
DATA CAPTURE 4.0/80 (VIDEX VER.)	89.95	56.00
WIZARDRY (SIR-TECH)	49.95	33.00
CASTLE WOLFENSTEIN (MUSE)	29.95	19.00
DEADLINE (INFOCOM)	49.95	31.00

ACCEPTED PAYMENT - PERSONAL CHECK or MONEY ORDER. NO CREDIT CARDS.
If Personal Check, Allow 10 BANKING DAYS for CLEARANCE.
Prices and specifications subject to change without notice.
Unless otherwise noted, ALL PRICES are PLUS SHIPPING.
SHIPPING & HANDLING — ADD $5.00 UPS SURFACE (DISK DRIVES $7.50 — MONITORS $10.00 — PRINTERS $15.00).
California residents add 6% Sales Tax.

Circle 12 on Reader Service Card

Speed,

Speed with the **PASCAL SPEEDUP SYSTEM,** our best seller. Features complete ease of use with all APPLE PASCAL; UCSD PASCAL 1.1; APPLE RUNTIME ENVIRONMENT; & FORTRAN applications. From 40 to 150% speed boost on most programs (some even faster). All with no programming and with only one extra watt of power from your power supply.

Power,

Power with the cost efficient 6809 coprocessor; the 8 bit chip with a 16 bit "personality". Enough power to drive multi-tasking OS-9 / BASIC 09, the "Unix-like" operating system with a Pascal-like BASIC for the Apple. It's all in a days work for **THE MILL.**

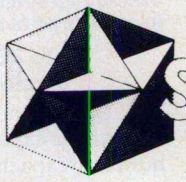
Efficiency.

Efficiency with the **ASSEMBLER DEVELOPMENT SYSTEM;** Heavyweights write state of the art programs on the 6809 for DOS 3.3 or even stand alone. Utilities include M.A.P. (McMill Adaptor for Pascal).

"The Stellation Two people have earned my highest respect for assembling an impressive collection of software for their board."
- Michael Coffey/
Creative Computing Magazine

"I feel my software output has been greater in the six months I have used OS-9 than it was in the 3½ years previous combined. It has a rakish logical simplicity that nearly defies description...I bill OS-9 as a programmers dream operating system."
- Brian Capouch/
MICRO Magazine

Call today or write for more information on this inexpensive Apple II, IIe enhancement.



The Lobero Building P.O. Box 2342
Santa Barbara, Ca. 93120
(805) 966-1140 Telex 658439

Circle 35 on Reader Service Card

TOMORROW'S APPLES TODAY

Kevin O. Donohoe/editor

NOTE: These products are presumed to be state-of-the-art. Call -A.P.P.L.E. does not necessarily endorse products published in this column as **we** have not tested them.

JOINT PROMOTIONAL AGREEMENT

Southwestern Data Systems announces a joint promotional agreement with THE SOURCE. For the first time, the superior communications software of "The Professional Series" has been combined with the highly acclaimed services of THE SOURCE. Effective immediately, each purchase of ASCII Express The Professional, P-Term The Professional, or Z-Term The Professional includes a subscription to THE SOURCE. Billed as "AMERICA'S INFORMATION UTILITY," THE SOURCE offers one of the most comprehensive data base and service selections in the industry. Luis Rivera, Vice President of Operations and Expansion for Southwestern Data Systems, expressed his intention to offer the most sophisticated and innovative communications software. Southwestern Data Systems, 10761 Woodside Avenue, Suite E, P.O. Box 582, Santee, California 92071, Telephone 619-562-3221.

KEYWIZ VIP

Creative Computer Peripherals, Inc., has recently announced its new auxiliary keyboard called the KeyWiz VIP. VIP stands for Very Intelligent Peripheral and for a good reason! It consists of 31 user programmable function keys housed within a new injection molded color coordinated case. Each of the 31 keys may be easily programmed with up to eight characters each and programmed again while using the shift key giving the user 62 user defined keys. Four such keyboards are stored in the devices memory, easily accessed anytime, making 248 preprogrammed keys available to the user at the touch of a button. A seven segment LED displays which one of the four keyboards you're using. Each key may be redesignated over and over again, even in the middle of a program as your needs change. It's independent of any software and can be ordered for the Apple][+, Apple //e, Franklin Ace or the TRS-80 Mod lil and others close behind. Creative Computers believes that its new products greatest asset is its ability to make all software *"User Friendly."* Creative Computers, Aztec Environmental Center, 1044 Lacey Road, Forked River, New Jersey 08731, 609-693-0002

HIGH SCHOOLS WALKMAN AND APPLE][

A computer accessory for secondary education health, science, athletics and computer education was announced by Axel M. Fritz, President, Vitalcomp Inc.

The accessory system, Fitness Logger, uses a Sony Walkman-type stereo recorder to tape students vital signs during regular activities, while participating in health or science laboratory classes, or during athletic activities on the field. The battery powered miniature ambulatory recorder then acts as a playback system feeding the body signal data into an Apple computer which has been programmed by an accompanying diskette, "A Very Personal History." The display shows the body signals in the standard ECG wave strip format along with the blood pressure wave form, each followed by the computed heart rate, the systolic, diastolic and mean arterial pressure (MAP). This data can then be manipulated as histograms, graphs, charts or other for comparison with past history in assessing fitness development. Each student may work with actual data from their own body and its' electrophysiological activity.

The Fitness Logger can also be used to record other low frequency signals with integral DC components such as EMG muscle signals, EEG brain waves or other electrical artifact. Direct play through the Apple][from the sensors without the recorder can be done with an optional optical isolator for real time cardiovascular review. An analog chart recorder can make hard copy for base line health records or perusal by health authorities. Details available from Vitalcomp Inc., Chanhassen, MN 553170175, 612-934-6166

NEW INTERFACE ADAPTS MOUSE TO ANY COMPUTER TERMINAL

A universal interface which allows a *Mouse* pointing device to be used with any computer terminal or personal work station has been developed by Random Access, Inc. The MU-2 Interface also is compatible with digital joy sticks, track balls and shaft encoders.

A high-resolutional digital input device, the *Mouse* moves the cursor on a computer display screen in direct proportion to its own movement. The Random Access interface with *Mouse* is inserted in the serial line between the host and terminal by means of a DB-25 connection.

The intelligent interface can be profiled from either the terminal or host. Direct, usable displacement data can be programmed to be sent to host or terminal. On-demand or on-motion transmission modes are profilable; transmission can be turned on or off under program control. A significant advantage is the system does not tie up additional host serial ports.

Data format can simulate cursor position control signals or cursor key commands. Data buttons can be programmed with up to four user-definable characters.

The Random Access MU-2 Interface is available with or without digital *Mouse.* The interface can be furnished as a component (a fully or partially stuffed PC board) for inclusion in other manufacturers' systems.

Additional details on the MU-2 Interface for *Mouse* may be obtained from Random Access, Inc., 246 Highland Road, Pittsburgh, PA 15235, 412-247-7472.

TOMORROW'S APPLES TODAY

AMPOWER INTRODUCES TRACK PEN®

Ampower Instrument Co., Inc. (NASDAQ:APOW) announced today the introduction of its newest product for the personal and industrial computer market — the TRACK PEN.

A hand-held device, the TRACK PEN, makes computers easy to use as it eliminates computer commands that have to be typed into the computer. With the TRACK PEN, which takes about 15 minutes to master, novices can organize and process data without touching the keyboard.

The TRACK PEN is similar in function, but provides the user with more speed, precision and dexterity than a "mouse," which is used in Apple Computer's new Lisa model. A "mouse" is a push-button, hand-held control device that is placed on a desk near the computer keyboard. By sliding the "mouse" on the desktop, an arrow points accordingly to a particular item in a "menu" of data-file arrays listed on the screen.

Unlike a "mouse," the TRACK PEN is ergonometrically designed to fit into the hand of the user, thus providing the user with more dexterity in manipulating data. The TRACK PEN is used like a pen on a desktop to track the cursor, a small white box on the screen of the terminal. The TRACK PEN, moreover, simplifies the operator's use of the computer by allowing the user to give an instruction to the Central Processing Unit by a "menu" which is displayed on the Cathode Ray Tube. The command is actuated by simply pressing the TRACK PEN's tip on the work surface area, after tracking a function on the "menu." Many functions within the program can be looked at quickly using the TRACK PEN.

Ampower's TRACK PEN, which will be marketed to original equipment manufacturers as well as end-users, is expected to retail for approximately $295.

Ampower Instrument Co., Inc., 26 Just Road, Fairfield, NJ 07006, 201-227-7720.

HOME HEALTH-DISK

Home Health-Disk is a newly released medical information system. It is well researched, consumer oriented, and attempts to increase your ability to care for yourself and manage your own health. It is a large program, two full disks long, operates on a 48K Apple][+, DOS 3.3, with one disk drive. It is available for $79.95 plus $1.50 for shipping and handling and comes with a three week money back guarantee. Contact Brahman Software Inc., P.O. Box 331, Castleton, Vermont 05735. Visa/Master Charge accepted. Phone orders to 802-775-2533.

IMPROVED SOFTWARE MANUALS

Is your software documentation scaring away potential users? Working from the premise that technical information can be taught simply. Technology Training Systems has developed 25 research-based guidelines for improving user manuals. Discover the practical and highly effective tips packed into our publication *24 Ways to Improve Your Software User Manuals.* This 15 page booklet can give your software the competitive edge and save time spent on user consultation and training. Technology Training Systems can help make your documentation "user friendly" through our critique, rewrite, development, beta testing, and training services. Your software documentation can help users:

- overcome technophobia
- learn quickly
- find information easily
- remember the essentials
- apply what's learned
- feel competent

To order a booklet and receive a summary of services, send $5.00 check or money order to: Technology Training Systems, 1078 Ravine Ridge Drive, Worthington, Ohio 43085. For a *free* IQ (Instructional Quality) CRITIQUE of one of your manuals, send a copy of your manual along with this announcement before June 30, 1983. For more information, call Kay Adams at 614-431-2174.

APPLECARE™

Apple Computer has introduced AppleCare™ Carry-In Service, a plan that lets dealers continue to provide quality care for Apple personal computers and peripherals under a single, comprehensive agreement.

For a fixed cost per year, the AppleCare Carry-In Service plan protects owners of Apple equipment against unexpected service expenses. Replacing the Apple Extended Warranty, AppleCare Carry-In Service extends the protection of the original 90-day warranty for an additional 12 months, and is renewable year after year. The plan covers parts and labor and offers unlimited repairs on all covered components.

With AppleCare Carry-In Service, owners purchase protection for individual components. The cost is less than 3/4 of 1 percent per month of the price of a typical system — Apple //e, Disk Drive II, and Monitor III. Under this plan, a typical system can be protected for a suggested retail price of $12.75 per month. The cost varies depending on the types of components covered. In most cases, this is lower than the Extended Warranty. Owners can purchase the plan at any time, and components can be added at a prorated price for the remaining time on an existing AppleCare Carry-In Service agreement. Available at your authorized Apple dealer.

TWO NEW PROFESSIONAL SERVICES ADDED TO SSM'S TRANSPAKS™

SSM Microcomputer Products, Inc. has included special introductory subscription offers to two more professional services — Dow Jones News Retrieval Service® and Dialog® Information Services. SSM's TransPaks now Include The Source™ subscription offer, one hour free of Dow Jones, and two hours free of Dialog through June, 1983.

Dow Jones News Retrieval Service provides information of business and financial news, current and historical stock quotes, detailed corporate data, general news, and the latest sports and weather reports from the publisher of The Wall Street Journal, Barrons and the Dow Jones News Service ("tickler").

The Source, Americas Information Utility, provides over 750 programs and services including electronic mail, travel services, newswire access, research data, discount shopping, games, stock reports, and investment analysis.

SSM is including all of these capabilities with their TransPaks at no additional cost to the user. The total value exceeds $200.

SSM Microcomputer Products, Inc., founded in 1974, manufacturers a wide range of interface cards and software for the Apple][and //e, as well as a complete line of board-level product for the IEEE-696/S-100 bus. SSM Microcomputer Products Inc., 2190 Paragon Drive, San Jose, CA 95131, 408-946-7400.

A NEW TESTING CONCEPT FOR THE APPLE][

There is a new way available to diagnose and troubleshoot the Apple][Computer — functional testing and exercising through control of the microprocessor (μP) bus. The concept of *bus control* involves replacing the Unit Under Test (UUT) μP with a test instrument that controls the UUT system μP bus, thereby allowing the technician to directly control the UUT system. This concept is used by the Fluke 9010A Micro-System Troubleshooter.

The Fluke 9010A Micro-System Troubleshooter

The troubleshooter is a new approach to μP-system troubleshooting. The 6502 processor in the Apple][is replaced by a Fluke Interface Pod, which is connected to the 9010A. This gives the technician total control over the processor bus, through the 9010A, thus allowing exercise of virtually every section of the Apple][circuitry.

Built-In Tests

Once the equipment is connected to the Apple through the micro-processor socket, the technician can manipulate the bus, using the wide variety of keyboard functions provided on the test device. A list of these functions is given in Table 1. In addition to these, the 9010A even has a Learn function, which can provide the technician with valuable information about the memory map of an unknown computer.

Functional Tests	Troubleshooting Functions
Bus Test	Read
	Write
RAM Short Test	Ramp
RAM Long Test	Walk
	Toggle Data
ROM Test	Toggle Address
I/O Test	Read Status
	Write Control
Auto Test	Toggle Data Control

Table 1. 9010A Keyboard Functions

Even when the Apple is crippled to the extent that the 6502 cannot operate, the 9010A gives the technician error indications as shown in Table 2. These messages give enough information to begin isolation of faults. Another "stopped-dead" situation is when signal lines on the bus are shorted together or held high or low. The Bus Test quickly checks that each of these lines can be toggled independently and, in case of a fault, indicates which lines are involved.

Error Message	Condition
BAD POWER SUPPLY	At least one supply voltage is bad
POD TIMEOUT	No clock signal
UUT POWER FAIL	No clock and bad supply voltage

Table 2. Catastrophic Error Messages

The most obvious things to do with the Apple's RAM are reading and writing. This can be done on individual bytes, of course, but there is a built-in RAM SHORT Test on the Fluke 9010A that makes block testing of RAM easy. This test checks that all bits are read-writable, checks that no lines are tied together, and does a statistical decoding test that is designed to catch most address decoding errors. Testing a 1K-byte section of the Apple's RAM takes about 13 seconds.

There is also a more exhaustive RAM test, the RAM LONG Test, available on the 9010A. In addition to everything that the RAM SHORT Test does, it also does a thorough decoding test, and a complete pattern-sensitivity test that guarantees, for any two bits, to set them to opposite states at some time during the test. The RAM LONG Test over a 1K-byte segment takes about 3.5 minutes. The people at A.P.P.L.E. have found the RAM Tests to be of great value, especially in isolating problems such as the failure of one bit in one byte of RAM.

If you choose to loop on either of these tests, you can use the 9010A to check, unattended, a segment of memory and stop when it finally encounters an error. This is great for those intermittent "flaky" RAM problems.

The ROM Test allows the technician to check the ROMs against known-good signatures. You can easily make a catalog of the signatures of the most common of the "non-standard" Apple][ROMs, too.

The rest of the Apple][(keyboard input, audio speaker, video mode selection, and external I/O) can also be checked by using reads or writes to the related "special" addresses.

Programmable

The 9010A also provides full programming capability of any sequence of keyboard operations, with the addition of labels and IF...THEN, GOTO, and DISPlay statements. Register operations such as AND, OR, INCRement, DECRement, shift left and right, and 1's COMPLement are also provided, allowing manipulation of data under program control. Any program developed, along with the run-time environment, can be saved on a microcassette tape.

An Apple][test tape has been developed at Fluke and can be obtained through your local sales representative from the John Fluke Mfg. Co. A copy of this tape has been in use at A.P.P.L.E. for servicing in-house Apple][']s. They have found it to be instrumental in locating the cause of some rather obscure failures.

The bus control testing concept provides a unique way to get "inside" the Apple, without having to write extensive software. This method also doesn't make any assumptions about a good μP, good bootstrap ROMs, or even good system RAM. Operation of the Fluke 9010A Micro-System Troubleshooter, in particular, is so simple as to make bus-control testing a very natural procedure.

Editor's Notes:

We'd like to hear from those of you who are using the Fluke 9010A for troubleshooting the Apple][Computer so we might share methods and 9010A software. Apple Puget Sound Program Library Exchange uses the Fluke 9010A for troubleshooting all office and some members Apples. Questions about the 9010A may be directed to the A.P.P.L.E. Office (technical support) or John Fluke Manufacturing Co.

FAMILY COMPUTING

FAMILY COMPUTING, the first national consumer magazine aimed specifically at families with home computers, will be published by Scholastic Inc. beginning September 1983, it was announced by Richard Robinson, president.

FAMILY COMPUTER, a monthly, is the first major consumer magazine to be published by Scholastic, a company that for over six decades has published high-quality, timely educational materials for all areas of learning for both schools and the home.

FAMILY COMPUTING will have an initial circulation of 200,000 paid, with a 200,000 bonus circulation. It will be aimed at families with children between the ages of 5 and 15 who have — or are contemplating the purchase of — a home computer. The black & white page rate will be $5000. Single copy price will be $1.95, with a yearly subscription price of $17.97 or two years for $27.97, and $29.97 for an annual foreign subscription. A dummy of the first issue will be available by early April and advertising will be sold at that point. Scholastic Inc., 730 Broadway, NY, NY 10003, 212-505-3000.

IMAGINATOR™

PROFESSIONAL 3D GRAPHICS PROGRAMS

Imaginator™ is an exciting new series of professional quality 3D graphics programs. Imaginator will let anyone interested in computer graphics create, edit, manipulate and display 3D graphics images. You do not have to be a programmer to use Imaginator. Three dimensional objects can be created without drudgery using Imaginator's powerful graphics editor. Once the objects have been created they can be manipulated and displayed in a variety of ways using Imaginator's unique and highly versatile set of display commands. Both novices and experts alike will appreciate Imaginator's extremely complete manual. The tutorial section of the manual not only teaches you how to use Imaginator; it also provides an excellent introduction to the subject to three dimensional computer graphics.

Imaginator runs on Apple® II, Apple II+ and the super new IIe computers (48K required). Works directly with Epson printers with Graftrax. Compatible with many screen dump utilities and graphics screen enhancers.

IMAGINATOR I An entry level program designed to provide the user with all the basic capabilities for creating, manipulating and displaying three dimensional objects. Perspective or orthographics views of an object can be displayed on the screen and then dumped to a graphics printer or saved to disk. (Fully upgradeable to Imaginator II for an additional $40). $79

IMAGINATOR II A professional level program allowing the user to create scenes with multiple objects. Makes full use of Imaginator's powerful graphics editor enabling the user to manipulate individual objects in a scene. Perspective or orthographic views of a scene can be displayed on the screen and then dumped to a graphic printer, saved to disk or plotted directly on a plotter. (Available third Quarter 1983). $119

Imaginator is a trademark of Shelter Research Institute, Inc. Apple is a registered trademark of Apple Computer, Inc. Epson & Graftrax are trademarks of Epson America, Inc.

TOWNSEND MICROWARE

A DIVISION OF SHELTER RESEARCH INSTITUTE, INC.

VISA 921 WATER STREET, P.O. BOX 1200, DEPT. 100B MasterCard
PORT TOWNSEND, WASH. 98368. 206-385-4621

Circle 26 on Reader Service Card

TOMORROW'S

FORTH-79 VER. 2 FOR Z-80 CP/M 6 APPLE USERS

MicroMotion has announced an expanded line of formats available for Z-80 1.4 & 2.x users. These include Apple, Micropolis Mod II, Vector Graphics Micropolis & Tandon, NorthStar, Cromemco, Health/Zenith, Osborne I, Kaypro II, Xerox 820, and TRS-80 Model II.

FORTH-79 is a complete professional software system that meets all provisions of the FORTH-79 Standard and standard programs are portable. FORTH-79 provides an interactive environment that allows rapid program development. Its high-speed compiled code makes it especially suitable for real-time applications. Programs are more compact and run faster than in BASIC.

Base system includes a screen editor, macro-assembler, string-package, three-bit integer arithmetic and 200 page tutorial and reference manual. Floating Point available for all version, High-Res for Apple & NorthStar. $99.95-$139.95. Contact MicroMotion, 12077 Wilshire Blvd., #507, Los Angeles, CA 90025, 213-821-4340.

A.P.P.L.E. Members Only

Low Price $60.00

NEW VERSION!

THE Spreadsheet

VERSION 2.0

The "NEW" generation Spreadsheet calculator program with all features necessary for creation of complicated models, forecasts and sales computations.

Supports

- Most popular 80-column boards.
- Optional software 70 column display.
- All "language card" compatible RAM cards (up to 512K).
- Full upper and lower case from any Apple computer.
- Variable column widths.
- Protected cells, hidden cells, and other advanced features.
- Field to field tabbing, for easy entry.
- Hard disk compatibility.

A.P.P.L.E. Orders
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245

- Washington state residents add 7.9% sales tax.
- Foreign shipping weight, 1 lb, 9 oz.

WRITING THE WRONGS

the wrongdoers

A Bug in LTXT

In *Call -A.P.P.L.E.*, September 1982, I described a new DOS command LTXT, for fast loading of text files. No bugs have yet surfaced in LTXT itself, but Ken Kashmarek reported a disconcerting side effect. He found that when LTXT was in memory, FID and certain other utilities were unable to write to a disk without clobbering its DOS image. Not only did the disk become unbootable, but the file being written was lost. The former disaster could be corrected with the help of MASTER CREATE to put a new DOS image on the disk.

When FID writes to disk it does so by the unusual method of a call to the DOS file manager, vectored through a jump at $3D6. When (and only when) called by this route the file manager makes use of data at $A909 to allow it to create new files. In my LTXT routine I had purposely altered this location to prevent LTXT from creating "non-existent" text files in the annoying way that OPEN does. On balance, the lesser evil seems to be to restore the byte to its old value, so that FID and the Toolkit assembler can work normally. One change is required to the original listing of LTXT, whose line 120 should read:

```
120 T$ = T$ + " NA909:21 7F":
    REM ALLOW PARMS A,L,B,R,
    S,D,V; ALLOW NEW-FILE
    CREATION
```

In the September 1982 article, I mentioned my dislike of altering DOS, and suggested that t' INIT command of any nonstandard DOS should be disabled to prevent propagation from disk to disk. Correction of the LTXT bug requires only a change in a program. There is no need to hunt through an indeterminate number of disk fixing up their DOS images.

R.D. Purves

&PEEK/&POKE Correction

There is an error in &PEEK/&POKE (Taking A Bigger Byte, February 1983, page 27). When &PEEK returns an address value, it fails to check that the variable is numeric or floating point. The variable can be checked for numeric by inserting the following statement before line #90 (TAX):

```
        JSR  CHKNUM ;$DD6A,
                    check for
                    numeric
```

The floating point check is inserted before line #91 (JSR MOVEMF):

```
        LDA  INTFLG ;$12 for
                    integer
                    test
        BMI  ERR0   ;branch if
                    integer
```

The error branch is inserted before line #93 (HIVAL):

```
ERR0    JMP  MMCH   ;$DD76,
                    mismatch
                    error
```

And the following definitions can be added at the beginning of the routine:

```
CHKNUM  EQU $DD6A   ;check for
                    numeric
INTFLG  EQU $12     ;integer
                    flat
MMCH    EQU $DD76   ;mismatch
                    error
                    message
```

This should avoid any problems if a character string variable or integer variable is used as the output for &PEEK (second argument). My thanks to C. Bongers for pointing this out.

Ken Kashmarek

Apple //e Monitor

Just a note to let you know that in disassembling the new Apple //e Monitor, I found that I had disassembled it incorrectly at location $FBB4 and subsequently identified the following source:

```
$FBB4: 06 08  ASL $08 ;Wrong
                      interpretation

$FBB3: 06     ???     ;Apple
                      Identification
                      Byte
$FBB4: 08     PHP
```

The bottom line is the only new zero page location used by the new Monitor is location "$1F". Location "$08" is not used!

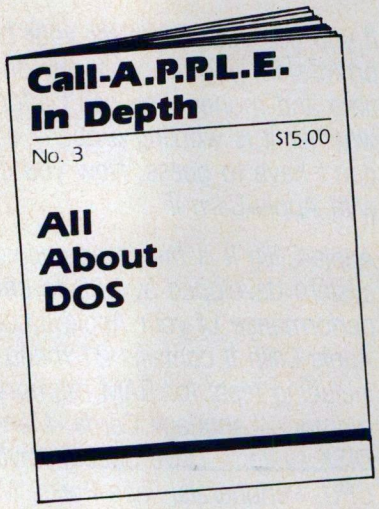
Art Schumer

ANNOUNCING!

A.P.P.L.E. Member Price

$13.50

ALL ABOUT DOS



288 pages of DOS tutorials and utility programs.

Non-Member Price: $15.00 (Available at your local computer store)

Dealer Inquiries Invited

A.P.P.L.E. Orders
21246 - 68th Ave. S.
Kent, WA 98032
(206) 872-2245

Wash. St. residents add 7.9% sales tax.

Overseas shipping wt. 2 lbs.

What's eating your Apple®?

Find out with Apple-Cillin II™

If you use your Apple for your business or profession, you probably rely on it to save you time and money. You can't afford to guess whether it is working properly or not. Now you don't have to guess. Now you can find out with Apple-Cillin II.

Apple-Cillin II is the comprehensive diagnostic system developed by XPS to check the performance of your Apple II computer system. Apple-Cillin II contains 21 menu driven utilities including tests for RAM memory, ROM memory, Language Cards, Memory Cards, DISK system, Drive Speed, Keyboard, Printer, CPU, Peripherals, Tape Ports, Monitors and more. These tests will thoroughly test the operation of your Apple, and either identify a specific problem area or give your system a clean bill of health. You can even log the test results to your printer for a permanent record.

Apple-Cillin II works with any 48K Apple system equipped with one or more disk drives.

To order Apple-Cillin II and to receive information about our other products, call XPS Toll-Free: ***1-800-233-7512.***

In Pennsylvania:
1-717-243-5373

Apple-Cillin II: $49.95. PA residents add 6% State Sales Tax.

XPS

XPS, Inc.
323 York Road
Carlisle, PA 17013

800-233-7512
717-243-5373

Apple is a registered trademark of Apple Computer Inc.

Circle 41 on Reader Service Card

The Pizza Program

Have you ever come home and wondered, "What are we having for dinner?" Now there's a great time saver for anyone who cooks or shops at the grocery store. Announcing the first dinner menu planning system.

ENDS HO-HUM DINNERS

Are you tired of the same old thing for dinner? Would you like more variety in your evening meal? Is there something you'd rather have but don't get very often? The Pizza Program is designed just for you. It's a delightful new software package designed to end the dinner-blahs with computer generated menus. Here is how it works.

You review what you like from the pre-selected food groups in the system. Delete any foods you don't enjoy. Add anything new at any time. Then decide how often you like to eat certain items. For example. Don't like liver? Then eliminate it with a few simple keystrokes. Or, you can plan for it as seldom as once every 99 weeks or as often as daily.

Want to go out to your favorite restaurant? Enter the restaurant's name as a "Main Course." Now your computer will automatically remind you to go out to eat—and as often as you select. It will delete all other items from that meal except the name of the restaurant.

AUTOMATIC SHOPPING LIST

You get a new menu each week or for just a few days if you want. And, it generates a detailed shopping list, automatically. It can arrange each item on the list in sequence according to the aisles at your favorite store. Studies show a shopping list will discourage impulse buying and save you money.

Also, it generates a per serving calorie counter. This is easy to delete anytime you are not in a diet mood or want to celebrate for any reason. You never count calories unless you want to.

RANDOMLY DELICIOUS

Say goodby to boring meals. Your computer will remember variety is the spice of life. This system makes eating at home a pleasure again. Each menu is randomly generated from major food groups according to the specific criteria you select. The system is easy to learn and easy to operate. Yet it is a sophisticated piece of software which will prevent menu mix-ups.

The Pizza Program is not a recipe file. Instead, it is a complete menu planning system you customize to fit your tastes and budget. It's a practical and easy way to organize your meal planning. You'll appreciate it day after day, week after week. People across the country are finding it a super idea—a great companion to any kitchen.

TRY IT FOR 30 DAYS WITHOUT RISK

You can try it with no obligation. Our home trial lets you actually use The Pizza Program for a full 30 days before you decide to keep it. Watch the fun and convenience it creates. Enjoy better meals and see how much time it saves. If you are not satisfied for any reason, return it within 1 month for a prompt and courteous refund. Your investment is just $34.50 plus $2.00 for shipping and handling. (California residents add 6½% sales tax.) A full 40 pages of instructions are included. And, it will be shipped promptly by first class mail.

OUT OF THE RUT

One woman's reaction to this program is typical. She wrote, "Before using your system I found myself getting into a rut of serving the same things over and over. The Pizza Program has changed all of this for me. We now have a wider variety of dinners and best of all I don't have to decide what they will be. If this was all it did, I'd be thrilled. But it isn't. The shopping list I receive along with my menus has been such a time saver. I quickly run through it and delete anything I feel I don't need and add something I might. I would have a hard time going back to doing my menus by hand."

This system requires an Apple II Plus* with 48K and 1 disk drive. (A printer is optional). We urge you to take advantage of our no-risk, 30 day home trial offer. To order call toll free and use your VISA or MasterCard. Or, send a check to the address below. There's no obligation. Order today.

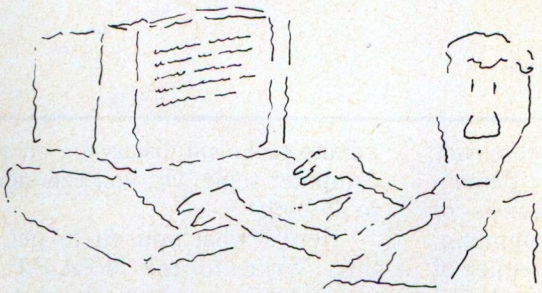
ORDER TOLL FREE
(24/hrs/day)
800-453-4000
(In Utah call **1-800-662-8666**)

Gourmet Software

671 Eden Avenue
San Jose, CA 95117

*Apple is a registered trademark of Apple Computer, Inc.

Circle 14 on Reader Service Card



Get a Grip on Machine Language.

The Visible Computer: 6502

The transition from Basic to machine language is a big step. Sometimes too big.

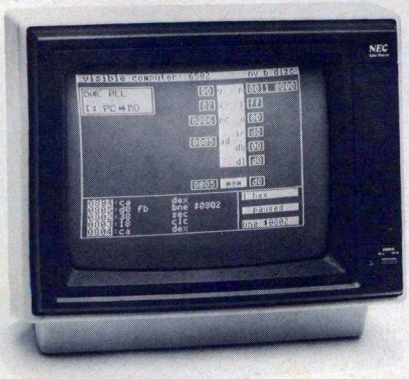
Now there's a way to cut the problem down to size.

The Visible Computer: 6502 is an integrated system for learning machine language programming. It features an animated simulation of the 6502 microprocessor that lets you see with your own eyes what machine language is all about. You'll see how instructions are executed, not just the result. There's a hand-holding mode that traps the kind of mistakes beginning machine language programmers are apt to make. And an expert mode, with more than twenty commands and functions that make it a powerful debugger for your machine language programs.

The 140 page manual is a hard-working tutorial on the fundamental concepts of machine language. You'll learn "hands-on-keyboard" as you work your way through thirty progressively more complex demonstration programs stored on disk.

The Visible Computer: 6502 is available for the Apple II Plus (or IIe) for $49.95. If your dealer doesn't have it, you can order directly from Software Masters, 3330 Hillcroft #BB, Houston, Texas 77057, or call (713) 266-5771. Please include $3.00 shipping. Bank cards accepted.

The Visible Computer lets you see into a 6502 as it executes programs



Software Masters™

Circle 38 on Reader Service Card

Apple® Owners save!

Prompt delivery and bottom dollar prices make Theta your single source for computer software. Listings represent only a portion of available programs. Call our convenient toll-free number for additional information.

[800] 233-7512 ext. 31

IN PENNSYLVANIA CALL:

[717] 243-5373

Business & Word Processing

ARTSCI	
Magic Window	73.65
Magic Window II	110.00
Magic Mailer	51.85
Magic Words	51.85
CONTINENTAL	
Home Accountant	55.38
DAKIN 5	
Business Bookkeeping	285.71
IUS	
Easy Writer	128.21
Easy Mailer	128.21
Datadex	110.00
LJK	
Letter Perfect	110.00
Data Perfect	85.81
MICROSOFT	
Multiplan	199.95
muMath/muSimp	182.73
Time Manager	110.00
TASC	128.21
Ramcard	75.90
Softcard	259.54
OMEGA	
Locksmith	75.90
Inspector	44.58
Watson	37.31
SILICON VALLEY	
Word Handler	145.66
List Handler	66.39
ON-LINE	
Screenwriter II Pro	146.35
The Dictionary	73.65
General Manager	110.00
Lisa 2.5	59.12
DISKETTES	
Elephant (SS/SD)	22.95
Elephant (SS/DD)	30.00
Wabash (SS/SD)	19.95
RIBBONS	
MX80 (6 pack)	29.56
NEC 55/77 Cloth (6 pack)	31.95
NEC 55/77 MS (6 pack)	25.78

Diversions

Choplifter	25.88	Frogger	26.40
Starblazer	24.22	Mouskattack	26.40
Prisoner II	24.96	Knights of Diamonds	26.40
Robot War	30.00	Beer Run	22.77
Castle Wolfenstein	22.77	Bandits	26.40

NO EXTRA CHARGE FOR MASTERCARD OR VISA. COD ORDERS ADD $3.00. ADD $3.00 FOR SHIPPING & HANDLING (INSIDE USA). PRICES SUBJECT TO CHANGE WITHOUT NOTICE.

Apple® is a registered trademark of Apple Computer Inc.

THETA

Research, Inc.

P.O. Box 140 • Carlisle, PA 17013

Circle 33 on Reader Service Card

A.P.P.L.E. INFO:

PACIFIC SCIENCE CENTER FESTIVAL

Michael Thyng/Call -A.P.P.L.E./staff writer

IT is time to sign up to participate in the *Second Annual Pacific Science Center and A.P.P.L.E.* Festival of Apple Computers for Computer Aficionados and the General Public.

Look. You know me from my articles in Applemash and the Robotics series. You know I wouldn't steer you wrong. Do something for yourself. Sign up to participate in this activity. Here's what happens.

On the weekend of June 11th and 12th, about 50 Apple computer owners will meet at the Pacific Science Center. There, we will set up our computers and talk with the general public about Apple computers. You will be able to tell the unknowing, first hand, why you chose the Apple and why they should choose the Apple. We will show the public and each other what we are doing with our Apples. Last year, we had people who used it for stock market analysis, game playing, preliminary psychological evaluations, demonstrations of LOGO, discussions of assembly language programming, business packages and a premier of an electronic magazine. This year I expect to have participants whose interest include all those things plus analytical graphics, technical discussions, *and* the public debut of the robot you've been reading about in *Call -A.P.P.L.E.*

This is one of the few times that you get to talk to the other computer people about what they are doing. Last year, our participants were really glad to have had a chance to get together. You know, we bill this as a benefit for the general public, but it is such a great thing for the participants that I have to admit it turns out to be mostly for them. There are two ways to participate in this. You can sign up to show what you are doing with your Apple. You can sign up to give a 20 minute technical discussion on some computer topic. Or, you can sign up to do both.

Here's what you do to get into this. Write to me % A.P.P.L.E. general office, and tell me what your area of major interest or specialization is. Tell me what computer equipment you will be bringing. Tell me if you want to give a technical talk. We will provide *no* hardware. We will provide electricity. But . . . do it today. If you want to give a technical talk, tell me soon so I can schedule you. Scheduling is usually a pain and late applicants will be given low priority.

Local computer stores will be invited to participate, and if you sell some kind of product for the Apple, you can sign up and bring it and show people. *But* under no circumstances are dollars allowed to change hands. (Pacific Science Center Policy). You can hand out circulars though.

TIRED OF PAYING BILLS?

Then let your Apple do it! Not just another check-book program, the BILL PAYER™ System turns your Apple into a complete check-writing and record-keeping system for all those bills that you have to pay on a regular basis. It powerful features include:

- A professional-looking check (check order forms included).
- All files & program on one disk.
- Fully prompting — easy to use.
- Maintains list of payees — pays bills QUICKLY!
- Has built-in reconciler — almost balances itself!
- Search capability
- Checks formatted for window envelopes — no more addressing!
- Disk is copyable for easy backup.

System requirements: Apple II, Applesoft, single disk drive, DOS 3.3, printer.

Send check or money order for **$49.95** to:

PLEASURE POINT ENTERPRISES
5483 PLEASURE POINT LANE
BELLEVUE, WA 98006

Washington State residents add $3.95 (7.9%) sales tax.

DEALER INQUIRES INVITED

Apple, Apple II, Applesoft, DOS 3.3 are trademarks of Apple Computer, Inc. Diversi-DOS is a trademark of DSR, Inc.

Circle 28 on Reader Service Card

219.95

FINALLY A RIGHT PRICE FOR THE RIGHT PRODUCT

micromax
presents
VIEWMAX-80

THE RIGHT 80-COLUMN VIDEO CARD FOR YOUR APPLE II*

It is more compatible with existing 80-column application software than other leading cards. And it is loaded with features. Check for yourself.

	RETAIL PRICE	7 X 9 CHAR DOT MATRIX	TRUE DESCENDERS	BUILT-IN SOFT VIDEO SWITCH	BUILT-IN INVERSE VIDEO CHAR.	FULLY PASCAL COMPATIBLE	SHIFT KEY SUPPORT
VIEWMAX-80	219	Yes	Yes	Yes	Yes	Yes	Yes
SMARTERM*	360	No	No	Yes	Yes	Yes	Yes
VIDEOTERM*	345	Yes	Yes	No	No	No	No
SUP'R TERM*	375	No	Yes	No	Yes	No	Yes
OMNIVISION*	295	No	No	No	Yes	No	Yes
VISION 80*	375	Yes	Yes	Yes	Yes	Yes	Yes
FULLVIEW 80*	379	Yes	Yes	Yes	Yes	Yes	Yes
WIZARD 80*	245	Yes	Yes	No	Yes	Yes	No

VIEWMAX-80 is functionally compatible with VIDEX's Videoterm and works easily with 80-column applications like CP/M, Pascal, WORDSTAR, AppleWriter II, Zardex, Easywriter, Letter Perfect, Magic Window, Apple PIE, VISICALC, ASCII EXPRESS, Transend, dBASE II, etc.

Shift Key Mod and Preboot disk for Visicalc and AppleWriter II available as low cost options. See your local Dealer or order direct. Visa/MasterCard accepted. Dealer & OEM inquiries welcome.

BACKED BY A FULL 2-YEAR WARRANTY.

DEALER & OEM INQUIRIES WELCOME.

micromax
6868 Nancy Ridge Drive
San Diego, CA 92121
(619) 457-3131

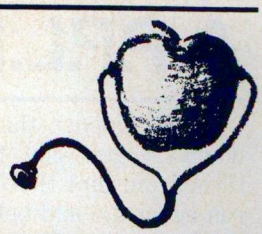
Also, Apple Parallel Printer Card; $79.95
Apple 40-Column Upper/Lowercase Adapter: $29.95
Z-80 CP/M Card: $149.95

"WE MAXIMIZE YOUR MICRO INVESTMENT"

*Trademarks of Apple Inc, Advanced Logic Sys., Videx Corp, M&R Enterprises, Computer Stop, Vista Computers, Bit 3 Computers, Wesper Microsystems.

Circle 1 on Reader Service Card

A.P.P.L.E. DOCTOR



David Sparks/Call -A.P.P.L.E. staff writer

Question:

EACH time I run a program that uses a random number generator I get the same sequence of numbers. That isn't very "RANDOM"! What's going on? How can I fix it so I will get truly random numbers?

Answer:

Computer-based random number functions were first developed for use in the study of statistics. Prior to the age of computers, statisticians in need of random numbers often looked them up in great, thick books filled with page after page of random values. When computers first became available to them, statisticians would hand-enter the tables, but they soon sought to compose computer programs that would save them the trouble.

Prior to the advent of computers, there was much debate among statistics types about how the books of random numbers ought best to be used. The computer did not solve this debate when it came along. Indeed, the computer made it possible to amass and test even larger bodies of purportedly "random" values. The debate grew as to what constitutes a "random" number.

Some things are pretty well agreed. To quote John Russ, et. al. in the January 1983 issue of *Call -A.P.P.L.E.*, "...the numbers themselves (must) be uniformly distributed over (a chosen interval, and) also the differences between successive values (and by extension, between any ordered pair) must also be random." In other words, as long as a large group of numbers produced by a method of calculation appears to be random when taken as a whole, the function is acceptable even though the method by which the values were obtained is quite systematic and non-random.

Indeed, in all such methods of calculation, or algorithms, known to us, each succeeding value is dependent upon the previous value, and the entire sequence is dependent upon an initial "seed" value provided to it by the programmer or the user. There is nothing at all random about the way "random" numbers are developed; the choice of the initial "seed" is completely arbitrary.

What you do not get from such a "random number generator" is a procession of numbers that are random in the way that dice are random. You do not get values that are *unpredictable from the outset, and totally independent from one another in succession,* as well as meeting other tests of randomness.

This, I believe, is where 99% of the confusion arises — a misunderstanding of what the words "random number generator" mean in a computer context.

The great change that Apples have brought to the world is to reverse the order in which people encounter computers and math. When computers were big, few, and expensive, access to them was limited to people who had "earned" it by virtue of mastering some part of math or science. They then learned to make use of the computer to further their studies at a fairly advanced level.

Today people who have little or no math background can own Apple computers. The writers of Applesoft, being sophisticated persons, saw fit to include a random number function. We who have little math sophistication observe that there is a random number generator at our disposal, and we may be excused for believing that it will do for us what a handful of dice would do — generate numbers at random.

A handful of dice is a random number generator, too. If you were patient enough, you could develop a huge list of values one at a time by picking up the dice, jiggling them, throwing them out, counting the spots, writing down the sum, and repeating the process many times. You would quickly tire of that, however. And a computer-based "random" number generator can produce a large list of such values that will serve the purposes of statistical analysis, such as simulating many plays of a given game of chance.

The dice also have the property of independence from one throw to the next. There is no way to foresee what the next throw will be, even if all previous throws are known. Here is a vital difference which suggests that dice are a "purer" and more "fair" source of random numbers when setting out to actually play a game of chance itself. (This assumes you are not playing with my cousin Smedly's dice, of course.)

We have never seen a computer program to simulate that second property of dice, independence of successive values, without access to some input from outside the computer. Most of the attention to random numbers in this journal thus far has focussed on the quality of large numbers of "random" values taken as a group. In this, the Applesoft RND function was found lacking. Crawford described the problem, John Russ et. al. suggested an explanation and proposed a replacement algorithm, USRND (Jan 1983), and your writer determined that RND did indeed contain the flaw which Russ had surmised. The replacement algorithm appears to solve RND's biggest problem.

Consider the example of a teacher writing a program to drill students on the simple addition facts. Using RND, the teacher would set two variables, A and B equal to values 1 - 9. The values would be displayed and the pupil asked to type the answer. Two problems eventually become apparent.

Suppose each pupil in the class gets to do twenty problems. After a time, some of the digits between 1 and 9 may no longer show up. Later pupils get incomplete drills. That is the problem USRND was written to overcome.

Now suppose that the pupils sit down the next day, the computer is turned on, and the drill program is run again. Lo and behold, here come the same examples in the same sequence as the day before. Third day, same thing. This is what has our questioner wondering. The answer is to change the initial "seed" value. For any given "seed" (and this is the key) the sequence of values which follow is fixed.

Since Applesoft always initializes the random number "seed" to the same value, our drill program always has to work with the same sequence. Unless, of course, we arrange to substitute a different "seed." To seed the Applesoft RND function, call it with the "seed" value expressed as a negative number. Example: X = RND(-N) where N is the seed value. Each time a given value of N is used, the same sequence of "random" values will follow. Page 102 of the *Applesoft Reference Manual* gives the details.

Where do you get "seed" values? This can be done in a number of ways. Asking the user to type in some number, then using that number as a "seed" is common.

We like to use the example of a bridge-playing program. It uses a random-number algorithm to "shuffle" the cards. For any given "seed," we get a specific sequence of "hands." Always the same sequence each time the same "seed" is provided. A different sequence for each "seed." We can decide whether to replay the same set of hands, or try a different one. You can see that having "random" sequences be dependent upon a "seed" value can be useful.

Back to our addition drill program. A canney pupil might discover that he can always get the same problems by giving the same "seed" value to the computer. Not good. Is there some way to simulate the independence of dice in the Apple? Turns out that there is. And from it can come a way to "seed" a random number function with a value that is unknown to the user.

Whenever the Apple is waiting for user input, it rapidly increments two memory locations, 78 and 79. Try this program:

```
10 INPUT "HIT RETURN: ";X$
20 PRINT PEEK(78):PRINT
30 GOTO 10
```

Demonstrate to your satisfaction that succeeding values cannot be predicted. Even if your "rhythm" is very good, the incrementing happens so fast that you cannot tap out a repeatable stream of values.

The value of either location will never exceed 255. Use modular arithmetic to confine the values to a smaller range. Change line 20 to:

```
20 X = PEEK(78):X = X - INT(X/9)
   * 9 + 1 : PRINT X
```

Try the same thing with location 79. It increments more slowly than location 78. In fact, only 1/256th as fast. Still pretty fast.

For larger values, try using PEEK(78) + PEEK(79) * 256. This will produce values ranging from 0 to 65535. The monitor is actually just counting up from zero to 65535 using these two locations. But it's doing it so fast that the built-in imprecision of human beings makes the actual value in the two locations unpredictable. That's what is meant by input from a source outside the computer.

A program could be written to produce an addition facts drill using this method. It would eliminate the problem of the fixed sequence. But it could be unwieldy for applications requiring more than a few dozen values, what with all the key pressing.

An INPUT request will have the effect of rapidly changing locations 78 and 79 regardless of what input is sought. Getting the pupil's name comes to mind. Locations 78 and/or 79 could then be used to "seed" a random function as follows:

```
15 X = RND(-PEEK(78)-1) : REM
   NOTE TWO MINUS SIGNS
   (NOTE ALSO THAT IT IS
   POSSIBLE FOR PEEK(78) OR
   PEEK(79) TO CONTAIN THE
   VALUE ZERO. ZERO IS NOT A
   VALID "SEED" FOR RND.
   THAT IS WHY THE "-1" IS
   INCLUDED IN LINE 15.)
```

That will place a "seed" value within the Applesoft RND function. If you are just using RND, knowing about "seeds" will help you.

If you are going one step further, and using USRND from the January issue of this magazine, "seeding" RND is the first step. From there, a further command to CALL 768 will cause USRND to convert the Applesoft "seed" into a "seed" for USRND. The completed "front end" for the addition facts drill using USRND might be:

```
5   PRINT CHR$(4);"BRUN
    USRND"
10  INPUT "TYPE YOUR NAME:
    ";N$
15  X = RND (-PEEK(78)) : CALL
    768
20  A = USR(9) + 1
25  B = USR(9) + 1
30  ANSWER = A + B
35  REM THE REST OF YOUR
    PROGRAM FOLLOWS
.
.
.
999 GOTO 20 : REM TWO NEW
    VALUES IN THE SEQUENCE
```

Note that you will surely be getting some input from the user during the course of this particular application. Therefore a GOTO 15 would reseed the function with some new, unpredictable value. In this way, you could doubly eliminate the problem of fixed sequence.

Question:

How do I use the new USRND function from machine language?

Answer:

I'm glad you asked. First of all, I will assume that you are using an assembler, such as Big Mac. I will also assume that you have the source code typed in from the January 1983 issue and have saved it to disk. I will refer to some labels and line numbers within the published listing. You might want to have it front of you now.

You will need to append the USRND code to your program. Big Mac has a command for doing this. That way, USRND will be assembled with your other code, so that it will not need to be loaded at $2F4.

You need to initialize the "seed." For that, label the instruction presently found at line 61, "GETSEED." This will correspond with the LDY #4 code. In your main program, JSR GETSEED will initialize the generator.

You may, in fact, delete lines 53 through 60. They set up an ORG that would not be appropriate, and put information on zero page that would only be of use to Applesoft.

To get a new value, you need to "float" the range argument, call the function, then convert the result back to a hex integer. This can be done this way:

```
LDY #RANGE  ;A VALUE 0 - 255
JSR $E301   ;FLOATS VALUE IN
            Y REGISTER
JSR RANDOM  ;GENERATES
            VALUE
JSR $E6FB   ;RETURNS VALUE
            IN X REGISTER
```

That's all there is to it. Both index registers are disturbed in this process. Save them first, if necessary. Refer to Crossley's article on Applesoft internals in *Call -A.P.P.L.E. In Depth #1 All About Applesoft* for more details on floating point and integer math routines.

Question:

How do I use the Applesoft random number function from within an assembly language program?

Answer:

It's a bit more tricky. We wrote USRND to accept a range argument and return whole number values within that range. The Applesoft RND function does not accept a range argument, and always returns a decimal fraction less than 1. The caller must further process the output of RND, particularly if that caller deals with hexadecimal integers, as a machine-code program usually does.

A JSR $EFAE will "build" a random fraction in the floating point accumulator (bytes $9D through $A1). You need to then multiply that fraction by a range value, and take the integer of the product. That will yield for you a value from zero to one less than the range value you used.

You might want to try something like the program on this page.

Thanks to John Crossley for discovering all those Applesoft internal locations. Refer to *Call -A.P.P.L.E.* In Depth #1 for his article, and one by Eric Goez on Applesoft floating point variables.

```
:ASM
                      1     RANDOM   EQU  $EFAE
                      2     FAC      EQU  $9D
                      3     FMULT    EQU  $E97F
                      4     QINT     EQU  $EBF2
                      5     *
                      6     * Written in Big Mac format
                      7     * more or less
                      8     *
8000: 20 AE EF        9              JSR  RANDOM      ;Make new random fraction in FAC
8003: A9 0D           10             LDA  #RANGE      ;Low byte of RANGE address in A
8005: A0 80           11             LDY  #>RANGE     ;High byte of RANGE address in Y
8007: 20 7F E9        12             JSR  FMULT       ;Multiply FAC by RANGE
800A: 20 F2 EB        13             JSR  QINT        ;Take the greatest integer of the product
                      14    * At this point, the low byte of your result is in
                      15    * hexadecimal form in $A1. The next most significant
                      16    * byte is in $A0, and most significant byte is in $9F.
                      17    * This is backwards from the way most values are kept
                      18    * in memory, but that's how the integer routine, QINT
                      19    * leaves them. Now, here is a floating point
                      20    * range constant. Satisfy yourself that it
                      21    * represents the value 40.
                      22    *
800D: 86 A0 00
8010: 00 00           23    RANGE    DFB  $86,$A0,0,0,0
```

Powerful — Economical — Professional

Peripherals for your Apple II*

MasterCard VISA

A/D 12-Bit, 16 Channel $450.00 AD — 121602

- Simple Software Selection of Channels
- Range ±10, ±5, ±2.5, +5, +10
- High-Speed 25 μ Sec. Conversion
- Full Software Support — Disk or Prom
- Adjustable Bipolar Reference

Ultra Rom Board/Editor $190.00 APB — 102

- Powerful G.P.L.E. [Global Program Line Editor] built in
- Includes: Search and Replace — Insert and Delete — and much more
- Edit programs 2 to 5 times faster
- 32K of Bank Switchable ROM Space
- Totally Transparent - Remove or Reload with a few keystrokes - without disk!
- Extensive Macro Table eliminates multiple keystrokes for common operations
- Useful Ampersand [&] Utilities all in one place
- Applesoft* Extensions including "IF, THEN, ELSE"
- Support for other HOLLYWOOD HARDWARE Products in Rom

6842 Valjean Avenue, Van Nuys, California 91406 (213) 989-1204 *Trademarks of Apple Computer Inc.

Circle 2 on Reader Service Card

Why Your Apple Needs DAVID-DOS™

Your Apple is easier to use and runs much faster with this new, licensed by Apple, DOS 3.3 update.

DAVID-DOS (rhymes with moss), is rated AA, (top of its class), by the highly respected software review magazine PEELINGS II.

Novices and Pros

DAVID-DOS is licensed by programmers for inclusion in the software they sell. 30 out of 60 bought it on the spot when DAVID-DOS was demonstrated at the Original Apple Corps, UCLA. DAVID-DOS receives applause from Apple novices. That's why you need DAVID-DOS.

What's Going On

Every Apple II/IIe boots up with DOS 3.3 software.* DOS 3.3 is right there on the first three tracks of all your program disks. Your Apple is supervised constantly by DOS 3.3.

Easier and 5 Times Faster

Our Installation Program goes in and makes specific permanant improvements to DOS 3.3 on all your disks in a few seconds. (The programs on your disks are not touched). From then on your programs load up to 5 times faster. And you have new features that make your Apple easier to use.

*Exceptions are Pascal & CPM Operating Systems.

Apple II and Applesoft are trademarks of Apple Computer inc.

100 Sectors in 7 Seconds

Speed Load Applesoft, Integer & Binary 100 sector programs in 7 seconds. Tload Random and Sequential Text Files at the same speed with a simple command.

Automatic Card Support

Automatically supports an Integer or Applesoft Rom Card in Any slot (without configuring). Single key stroke Catalog and Catalog Abort. Shows Free Sectors Left on each Catalog.

Nine New DOS Commands

1. TLOAD speeds loads all Text Files.
2. TLIST lists all Random/Sequential Text Files.
3. DUMP Binary/Ascii to screen or printer.
4. DISA dissembles Binary to screen/printer.
5. AL prints program Address & Length.
6. / Single keystroke, second Catalog command.
7. HIDOS moves DOS to the Language Card.
8. FIND hex group in 64K memory in 3 seconds.
9. DATE prints out. Use with Mtn. clock card. Commands 8 & 9 in Hidos only.

10K More Memory

These nine commands operate identical to existing DOS commands. Use A or A$ for address and L or L$ for length. Enter new commands on the Keyboard and use them in Basic programs with the familiar D$. Use HIDOS in Hello program for turnkey startup, adding 10K free memory.

Variable Speed Scrolling

TLIST, DUMP, and DISA Features are:

1. Print with PR#.
2. Key operated variable speed control.
3. Instant pause with Space Bar.
4. Exit anytime with Return Key.

Compatible

All DOS entry addresses have been preserved. DOS is Same length and compatible with most software. David-Dos is copyable and creates fully copyable updated disks. Init areas were used for David-Dos. Works with all Apple IIs including IIe 80 Col. Requires 48K. Complete documentation for screen or printing and many utilities are on the disk.

California residents add 6½% sales tax. **$39.95** Foreign add $2.00.

DAVID DATA™

To Order: Send Check To
12021 WILSHIRE BLVD., SUITE 212A
LOS ANGELES, CA 90025 (213) 478-7865

Circle 13 on Reader Service Card

The Executive Secretary™

- **Document editor**
- **Powerful printing program**
- **Flexible database**
- **Alphabetical indexer for books and theses**
- **Mail list merge for business letters**
- **Pre-printed forms manager**
- **Electronic mail**

INTEGRATED means that all of the above is included in the $250 suggested retail price, and that every part of the package follows the same rules. There's no need to learn one set of keypresses for the editor and a separate set of rules for the database.

RELIABLE means that thousands of users have logged more than 1,000,000 hours of use and that we back you up after the sale with a hotline number printed right on the program disk.

INTEGRATED, RELIABLE WORD PROCESSING

THE EDITOR: Supports a wide variety of 80 column boards, or runs in 40 column mode on older Apples. A coupon for a free Revision 7 LCA is included.

THE PRINTER: You can embed format commands in your document and let our printer program handle all of your layout concerns. More than 40 embedded commands allow you to control every aspect of layout, including pagination, binding margins, flexible page headers, volume-length documents, outline indentation, and far more. You can employ electronic shorthand so that commonly used phrases are represented by three or fewer keystrokes. And if your printer is capable of it, you can do justification by incremental spacing, boldface, and more.

ELECTRONIC CARD FILE: With its ease of use, its multiple alphabetic and numeric sorts, its report options that include totals and subtotals, it is superior to many databases costing well over $100.00.

MAIL LIST MERGE: You can write documents in the editor that call for information from the Electronic Card File and merge the information to produce customized forms and form letters. Conditional printing allows you to skip words, blocks of text, or entire letters based on your criteria. And you can also draw information from other popular databases.

ALPHABETICAL INDEXER: This unique tool allows you to specify page numbers for each occurrence of a term and to produce a properly formatted alphabetical index of those terms.

PRE-PRINTED FORMS MANAGER: Use a sequence of special embedded commands to define the row and column positions of the blanks on pre-printed forms. Our template helps you. Then use mail-list merging to combine information from your database with your pre-printed forms.

ELECTRONIC MAIL: With a Hayes Micromodem you can transfer your documents to another computer—or to your local typesetter for printing.

ORDER BY PHONE

(612) 929-7104

Available at your local dealer.

SOF/SYS, INC.

4306 Upton Ave. So. • Mpls., MN 55410

Circle 44 on Reader Service Card

CONSULTANTS

John A. Hall, Jr.	Apple //e	(617) 241-9050
Bob Huelsdonk	Apple ///(Western U.S.)	(206) 362-4910
Stuart Yusem	Apple /// (Eastern U.S.)	(609) 822-9306
Neil Clemmons	Agriculture	(314) 968-9618
Erv Edge	Assembly (Alaska)	(907)337-7620
Dennis Baumgarte	Assembly (Eastern U.S.)	(914) 473-3210
Don Williams	Assembly (Western U.S.)	(206) 242-6807
Michael Thyng	Business	(206) 524-2744
Murray Arnow	CPM/Engineering Applications	(312) 291-9210
Steven E. Bair	Data Bases	(913) 782-5901
William P. Steinberg	DOS (Eastern U.S.)	(212) 921-0663
Art Schumer	DOS (Northern U.S.)	(701) 282-7907
Lee Meador	DOS (Southern U.S.)	(817) 469-6019
Kurt Thams	DOS/Gen'l (Western U.S.)	(415) 327-2116
WM. Reynolds III	DOS/Tech (Western U.S.)	(503) 472-4028
Ted Perry	Education	(916) 485-1690
Cornelis Bongers	European Consultant	010-525511 Ext. 3275
Alan Robinson	Fortran	(503) 757-8234
Jim Eatherly	Games	(202) 232-6046
Tedd Sperling	Geoscience	(517) 393-1296
Stan Axelrod	Hardware Design/I-O Control	(415) 488-0915
Tom Deal, M.D.	Handicapped Applications	(206) 634-2847
Randy Halvorson	Hardware Interfacing	(206) 783-9040
Ed So	Hi-Res	(408) 267-3448
Stowe Keller	Mathematics	(804) 973-7647
Fred Merchant, M.D.	Medical Group	(415) 921-7792
Stephen F. Lee	Music	(704) 289-4102
Tom Woteki	Pascal (Eastern U.S.)	(202) 547-0984
Paul E. Sleeper	Pascal (Western U.S.)	(415) 494-1915
Bruce E. Sargent	Pascal (Western U.S.)	(206) 488-1161
Randy Warrens	Pilot Language	(608) 837-9566
Hersch Pilloff	Stock Market	(301) 292-3100
Jim Hockenhull	Video Interfacing	(509) 334-4137
George L. Hefter	VisiCalc (East)	(617) 256-8148
Jerry Brieger	VisiCalc (West)	(206) 881-1241
Vince Aderente	VisiCalc (North)	(312) 289-9685
Dr. Bob Richard	VisiCalc (South)	(504) 466-0894
Brian Abrahams	VisiCalc (South Africa)	(Durban) 315442
Michael Ginsberg	Word Processing	(805) 688-1465

AD INDEX

A.P.P.L.E.	36, 70, 71
Anthro-Digital	60
Apogee, Inc.	77
Apple Computer	40, 41
Assembler Teacher	44
Beagle Brothers	34
Bytes & Pieces	49
Computer Case Co.	32
Dajax	49
Datamost	4
David-Data	78
Digital Acoustics	50
Gourmet Software	72
IJG, Inc.	55
Ivers Specialties	32
Laing Electronics	16, 66
Leading Edge Products	IFC, BC
Legend Industries	53
Madwest Software	38
Micro Max	74
Orange Micro	64, 65
Pacific Computers	16
Passport Design	5
Pleasure Point Software	74
Professional Data Systems	42
R.C. Electronics	18
Scientific Software	14
Sensible Software	63
Soft Key Publishing	38
Sof/Sys, Inc.	11, 78
Software Masters	73
Star Logic	46
Stellation Two	66
Tackaberry Software	44
Tava	26
Telephone Software Connection	25
Theta Research	73
Townsend Microware	70
Transtar	8, IBC
United Computer	6
Video Associates Lab	56
Vista Computer	1
XPS, Inc.	72

FOREIGN MEMBERS: Please calculate postage based on weights in product descriptions.
(See postage table below for cost)

FOREIGN SHIPPING CHART

POSTAL TABLE		I West Indies Central American Columbia, Venezuela	II All South American Countries (except in Table I), Europe, N Africa	III Asia, Mid East, S. Africa Australia and Pacific Islands
0 oz	1 oz	$0.50	$0.60	$0.70
1	2	.76	.96	1.16
2	3	1.02	1.32	1.62
3	4	1.28	1.68	2.08
4	6	1.54	2.14	2.74
6	8	1.80	2.60	3.40
8	10	2.06	3.06	4.06
10	12	2.32	3.53	4.72
12	14	2.58	3.98	5.38
14	16	2.84	4.44	6.04
16	18	3.10	4.90	6.70
18	20	3.36	5.36	7.36
20	22	3.62	5.82	8.02
22	24	3.88	6.28	8.68
24	26	4.14	6.74	9.34
26	28	4.40	7.20	10.00
28	30	4.66	7.66	10.66
30	32	4.92	8.12	11.32
2 lbs	2.5	5.96	9.96	13.96
2.5	3.0	7.00	11.80	16.60
3.0	3.5	8.04	13.64	19.24
3.5	4.0	9.08	15.48	21.88
Each additional ½ pound over 4 pounds add:		1.04	1.84	2.64

CANADA SHIPPING RATES

Up to:	
1	.20
2 oz	.37
3 oz	.54
4 oz	.71
5 oz	.88
6 oz	1.05
7 oz	1.22
8 oz	1.39
9 oz	1.56
10 oz	1.73
11 oz	1.90
12 oz	2.07
1 lb	2.58
2 lbs	3.10
3 lbs	4.10
4 lbs	5.10
5 lbs	6.10
6 lbs	7.10
7 lbs	8.10
8 lbs	9.10
9 lbs	10.10
10 lbs	11.10

For weights over 10 lbs, add $1.00 per pound.

ORDER FORM

Membership # _____ Expiration date _____ Date _____

Name _____ Address _____

City _____ State/Country _____ Zip _____ Phone _____

PRODUCT	Unit Price	*FOREIGN SHIPPING WT.	AMOUNT
1			
2			
3			
4			
5			
6			

Mail to: A.P.P.L.E.
21246 68th Ave. S.
Kent, WA 98032
(206) 872-2245
24 hr. toll free order number (800) 426-3667

VISA Account No. _____ Exp. Date _____

MasterCard Account No. _____ Exp. Date _____

Subtotal _____

**Washington sales tax _____

TOTAL AMOUNT _____

*Foreign Members Only
**Washington state residents add 7.9% sales tax.
Sorry, no foreign hardware orders.
Now a 24 hour toll-free order number for members in 49 states (Washington excluded) and Puerto Rico. Sorry, product information not available. **800-426-3667**

UNCLASSIFIEDS — A.P.P.L.E ORDERS

Call -A.P.P.L.E. (un)Classifieds

Members Non-commercial rate .05/word
Members commercial rate .25/word
All others: 1.00/word, $25 minimum
Send ad copy with payment to editorial office. Unclassified copy must be received by the first day of the month preceding the date of issue.

NO MEMBERSHIP FEE. Rent software $5 month; buy it, 50% discount; sell it, receive credit on purchases or rentals; swap it, call (206) 784-2081 or send SASE to: Preowned Software, 4229 Baker Ave. N.W., Seattle, WA 98107 for details.

WORD HANDLER, new, mint, not booted even once, complete $135.00. Used Silentype Printer and Interface, works perfectly, $150.00. Videx Lowercase Chip, never used, $15.00. Buy all, 10% off. D Lander, M.D., Rt. 4, Box 130, Floyd, VA 24091.

MICROSOFT Z-80 SOFTCARD, complete set, used once $189.00. APPLE PASCAL 1.1, original diskettes and manuals PLUS Pascal In Depth 2 diskette set $189.00. Both for $360.00. Shipping included. (619) 757-3513.

FOR SALE: Enhance your Apple] [with Apple //e features. VIDEX Keyboard Enhancer] [for $99 (Retail $149) offers lowercase letters and auto repeat keys plus many features not found on the Apple //e. VIDEX Videoterm Board for $199 (Retail $345) offers 80-column display. VIDEX Soft Video Switch for $19 (Retail $35). 16K RAM Board for only $49 (Retail $195). Excellent condition. Original cartons and documentation. Bob Britton, 6111 Jadecrest Ct., Spring, TX 77379, (713) 376-7525.

FOR SALE: Microsoft Z-80 card, complete documentation $200.00, Mountain Hardware ROM+ board $150.00, Epson MX-80 Printer (parallel) $275.00, Anadex DP 8000 Printer, working condition $125.00, 2 Apple Parallel printer cards with cables $50.00/ea, 2 Tymac Parallel printer cards $50.00/ea, Apple Silentype II Printer, still in box $280.00. Call (602) 979-4554 for additional information. Prices do not include shipping charges.

INTERACTIVE STRUCTURES PKASO EP12-80 PRINTER INTERFACE FOR EPSON MX-80, MX-100 — Never used, in original shipping box, $90. John Conrad, 121 N. Allen St., Madison, WI 53705, (608) 238-0595.

A MUST FOR APPLE USERS. The original Reminder Calendar. Better than an electronic appointment book, great for everyone from businessmen to housewives. You'll enjoy using it everyday. Send $24.95 to: "CALENDAR" P.O. Box 5701, Forest Pk. Branch, Dayton, Ohio 45405.

SILENTYPE 80 PRINTER. $175.00. (208) 664-0372.

DISCOVER THE CORE OF YOUR APPLE: Get the complete inside story on the core of your Apple, the 6502 microprocessor. You've programmed it, played with it and cursed at it, now discover the 6502's intimate details. Get your very one APPLE CORE KIT which includes a giant 400 square inch full color blowup of the 6502 microprocessor, plus the 6502 White Paper. Together they reveal the chip's history, its topology, a detailed pin description, the location and function of each section of silicon, as well as showing the incredible detail of the Apple's core enlarged over 20,000 times. Only $5.00 to: Omicron Digital Systems Ltd., P.O. Box 3055-ACK, Falls Church, VA 22043.

MICROSOFT MULTIPLAN ELECTRONIC WORKSHEET. Infoworld's "software of the year." Native or CP/M. Reg. $275. Special $199. Send money order or cashier's check for faster delivery to Maritime Software Associates, 6660 Reservoir Road, Corvallis, OR 97333. (503) 929-2552.

APPLE PASCAL SOFTWARE DEVELOPERS. Move up to the next generation. SAGE II. The 16-bit supermicro with the best price/performance ratio. Comes with Softech's IV.1 UCSD p-System code compatible with IV.0/IV.1 Apple PASCAL. Sage system includes UCSD PASCAL, FORTRAN 77, compiled BASIC, 68000 ASSEMBLER and Sage utilities; 128K RAM expandable to 512K; two 80-track drives (1280K). Special offering at $3300 (reg. $4400). For information contact Maritime Software Associates, 6660 Reservoir Road, Corvallis, OR 97333, (503) 929-2552.

APPLE SOFTWARE RENTAL & sales. Over 2500 programs available. Rent that expensive software for just pennies a day! Digital Delli Software. 2961 S. Las Palmas, Mesa, AZ 85202, (602) 820-0082.

APPLE PASCAL ANIMATION GRAPHICS Morwe/Graphics - 83 pp. manual and two diskettes. SASE for details, $3.95 for manual. **-$19.95-** for complete kit. HENDERSON ASSOCIATES, 980 Henderson Ave., Sunnyvale, CA 94086.

AIR NAVIGATION TRAINER. Features VOR's, NDB's, ADF, adjustable winds, ground-track map and more. Call or write for more info. SPACE-TIME ASSOCIATES, 20-39 Country Club Drive, Manchester, NH 03102, (603) 625-1094, MC and VISA, CL0393 and 70310,217 US$40.

AMATEUR RADIO OPERATORS! - The Radcom Plus+ Package consists of a quality TU interface that installs in the APPLE, connects to your rig, and uses SUPER-RATT, the most advanced, feature packed software every developed for SENDING/RECEIVING RTTY and MORSE CODE. Features like SELCALL, a RADIO BULLETIN BOARD SYSTEM, Verified File Transfer, Printer, Clock routines and much more. Detailed information from: ALEX M. MASSIMO - AF6W, 4041 41st. Street, San Diego, CA 92105.

COMPUTER ROAD ATLAS - Start your next trip with a computer listing showing the best route. ROADSEARCH computes and prints the shortest practical route with miles, time and gallons. Contains 406 USA/Canada cities and 69,000 road miles. DOS 3.3 Ten day MONEYBACK GUARANTEE. $34.95 (plus $1.50 shipping) Check/VISA/MC. Columbia Software, Box 2235M, Columbia, MD 21045. (301) 997-3100.

THE CHEAP ASSEMBLER. An APPLE assembler with: ten command text editor, two pass RAM/disk based assembly, free field programming, interactive operation, tutorial manual and demonstration routines on disk. APPLE] [+, 48K, DOS 3.3 required. Send $20.00 + $4.00 P&H to: Thunder Software, P.O. Box 31501, Houston, TX 77231.

SYM-1 Single Board Micro-8K Onboard RAM, BASIC in ROM & Extended Monitor Eprom. With all manuals. First $125. J. Wisnia (617) 272-4740 evenings.

SOFTALK'S BEST SELLERS at the lowest prices in town. The Country Store, Box 2131, Dept. CA, Littleton, Colorado 80161, (303) 779-0883. Write or phone for free price list.

VIDEOTERM OWNERS. New character set EPROMs now available: hi-light/lo-light set (great with WordStar); APL; square; bold face; several foreign languages. Only $19.95 each, hi-light/lo-light pair $34.95. Add $1.50 for shipping/handling. (Add 7.6% tax if Washington resident). Custom character sets available on a quotation basis. Write for literature including pictures of character sets. KJT Enterprises, P.O. Box 762, Mercer Island, WA 98040.

GOLDCOAST COMPUTER RENTAL & SALES. 3000 programs available for rent or sale. "Rent to buy - but try before you buy." Software, firmware, hardware, publications. Goldcoast Computer, 2014 Triplett St., Owensboro, KY 42302.

(Continued on page 59)

PRODUCTS NOW AVAILABLE TO A.P.P.L.E. MEMBERS

Call —A.P.P.L.E. Specials

PEEKing Special[1]	$20.00
Annual Special[1]	$25.00

A.P.P.L.E. LIBRARY

"Apple Orchard" No. 1 (magazine)	$2.50
Apple Users Guide	$13.00
Assembly Lines: The Book	$17.00
Beneath the Apple DOS	$17.00
"Best of Cider Press" (S.F. Newsletter Annual) Vol. 1	$4.50
Call A.P.P.L.E. Annuals	
1978	$7.50
1979	$10.00
1980	$15.00
1981	$18.50
1982	$18.50
Call —A.P.P.L.E. in Depth	
1: All About Applesoft	$7.00
2: All About Pascal	$11.00
3: All About DOS	$13.50
Kids and the Apple	$16.00
Power of VisiCalc Vol 1 or II, each	$8.50
Using 6502 Assembly Language	$17.00
Woz Talks (VHS Video cassette)	$27.50

AS IS SOFTWARE

Public domain diskettes (Send 37¢ S.A.S.E. for list of titles)	ea. $4.00
Eamon Adventure Diskettes	ea. $4.00

A.P.P.L.E. DISKPAKS

Call —A.P.P.L.E. Anthology Vols I-VII, incl.ea.	$12.50
DICE 82[2]	$12.50
Diskpak 12	$16.50
Diskpak 34A	$13.00
Diskpak 5	$13.00
Diskpak 6B (DOS 3.2 disk utility)	$18.50
Diskpak 6B-16 (DOS 3.3 disk utility)	$18.50
Diskpak 7	$17.50
Diskpak 8	$13.00
In Depth 1 Diskette	$16.50
In Depth 2 Diskette Set	$20.00
Pascal Anthology '81	$12.50

A.P.P.L.E. HARDWARE

ApController	$85.00
Ap Coolkit	$20.00
AP DRIVE	$240.00
AP DRIVE with Controller	$325.00
Ap L Case 1	$30.00
Ap L Case 2 (Lower case chip)	$25.00
Disk Banks	$4.50

AP• Jewelry

Lapel pin	$3.00

A.P.P.L.E. PRESTIGE Game Diskettes

Cosmic Combat	$12.50
Mummy's Curse	$12.50

Apple Prestige Product LinE

APfile — Text file utility	$18.50
ApGraphics: The Filler	$17.00
Apmail][Version 3.0	$30.00
Applesoft Carpenter	$18.50
Aptest	$18.50
Big Mac Macro-Assembler/Ted	$22.50
Big Mac LC	$28.50
Global Program Line Editor	$38.50
Higher Fonts 1	$13.00
Higher Text Plus	$27.50
Program Global Editor	$18.50
Program Line Editor	$20.00
Soft Seventy	$20.00
Symbol Symon	$20.00
THE Spreadsheet Version 2.0	$60.00
WordWeaver ///	$75.00

[1] Expires June 30, 1983
[2] $7.50 With Ap L Case 1
$10.00 with Ap L Case 2

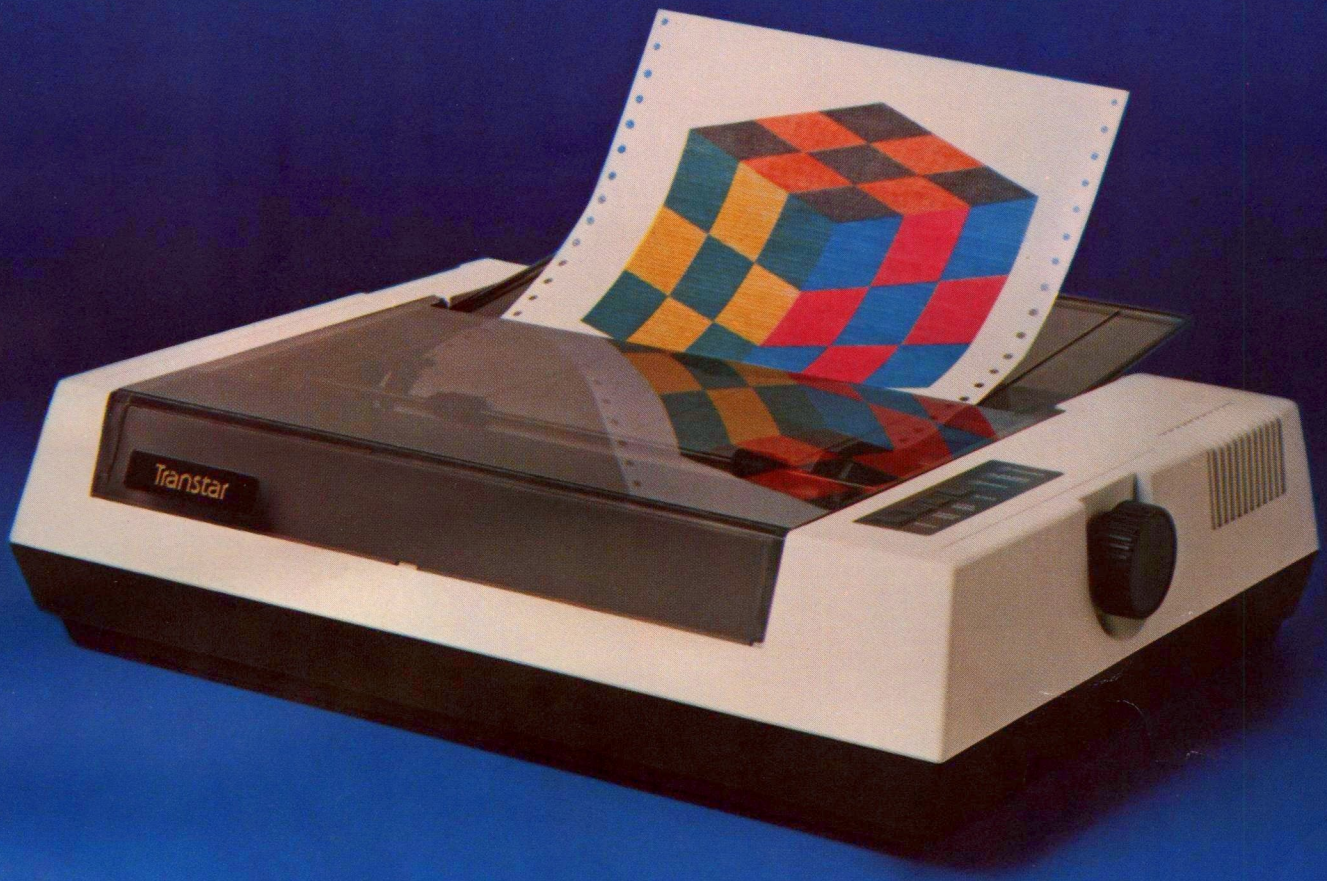
Diskettes supplied on DOS 3.3 except as noted.
All walkin orders must be called in prior to pickup.

NOTICE TO FOREIGN MEMBERS:
A.P.P.L.E. Hardware products not available to overseas or foreign members. This includes Ap.Drive & Ap Cool Kit.

Washington state residents add 7.9% state sales tax.

ORDER FORM ON REVERSE

Color for the Price of Black and White



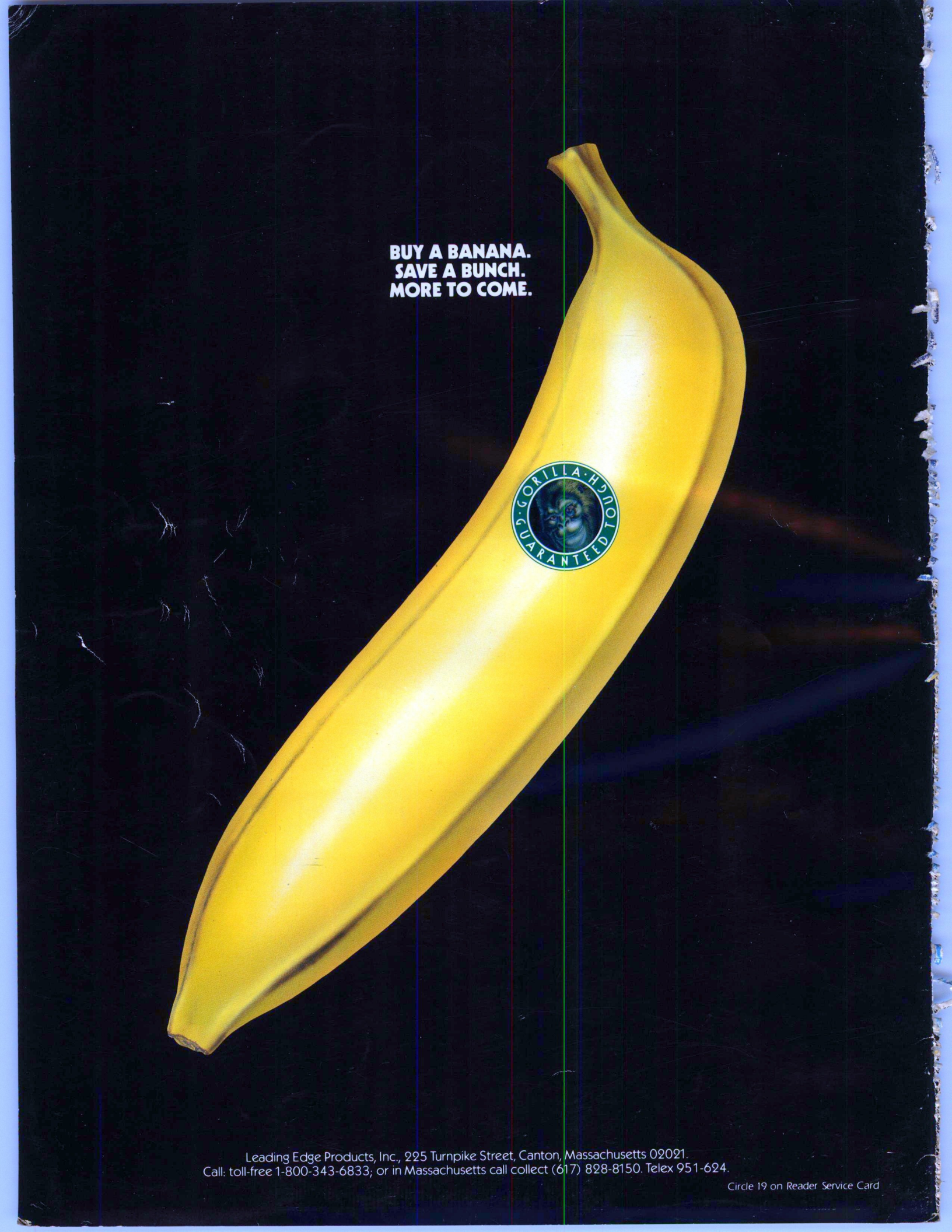
Transtar 315

Color Printer for only $599 suggested retail price

- New technology four-hammer print head
- Unique diagonal ribbon provides simpler and more reliable operation
- Prints 7 colors plus more than 30 shades
- All colors and shades are printed in a single pass of the print head
- Built by Seikosha, the oldest manufacturing company of the famous Seiko group—providing precision products since 1892.

Transtar · Box C-96975 Bellevue, Washington 98009

Circle 43 on Reader Service Card



Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021.
Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.

Circle 19 on Reader Service Card